Evaluation and Optimization of 2D/3D Image Registration in Robot-Based Minimally Invasive Spine Surgery

DISSERTATION

zur Erlangung des Grades eines Doktors

der Ingenieurwissenschaften

vorgelegt von

M. Sc. Julio Cesar Alvarez Gomez

eingereicht bei der Naturwissenschaftlich-Technischen Fakultät
der Universität Siegen
Siegen 2021

Betreuer und erster Gutachter

Prof. Dr.-Ing. Dr. h. c. Hubert Roth

Universität Siegen

Zweiter Gutachter

Prof. Dr.-Ing. habil. Marcin Grzegorzek

Universität zu Lübeck

Tag der mündlichen Prüfung

13.06.2022

Wei-Ping Hsu, you were my constant support during this adventure; unfortunately, you are no longer here to see how it ended. This thesis is dedicated to your loving memory.

Acknowledgments

First of all, I would like to thank my thesis advisor Prof. Dr.-Ing. Dr. h. c. Hubert Roth for his support during all this time. I want to thank Prof. Dr.-Ing. Habil. Marcin Grzegorzek for taking the time to be the second evaluator of my work. I would also like to thank Dr. Jürgen Wahrburg for his infinite patience and care over the last five years.

I would like to express my gratitude to my colleagues from the Zentrum für Sensorsysteme, Oliver Gieseler and Dr. Miguel Heredia, who contributed enormously to my research and gave me invaluable comments during the development of this work.

I want to acknowledge my colleagues of the RST department, Marc-Simon Schäfer, Peter Sahm, Omar Gamal, Nasser Gyagenda, Daniela Escobar, Erwin Gerz, Dominik Scarpin, and Jasper Valentine Hatilima, who were always there with valuable suggestions.

Finally, I want to thank my mother, who despite the distance, has shown me her support and understanding during these years.

Table of Contents

Abstract

Advances in medical imaging technologies have made possible minimally invasive surgery, which in comparison with conventional open surgery, leads to a faster procedure execution, reduced incisions sizes and bone exposure, and quicker recovery times. A particularly complex minimally invasive procedure is the insertion of pedicle screws in the lumbar spine, requiring accuracy grade A or B on the Gertzbein-Robbins scale, i.e., below 2mm. A minimally invasive surgical procedure starts with pre-operative imaging acquisition, e.g., a 3D-computed tomography (CT-scan), usually taken outside of the operating room (OR) and used to diagnose and plan implant insertion. A significant challenge consists of bringing the planned data into the OR, i.e., registering pre-operative data with the current patient pose using ubiquitous imaging modalities found in operating rooms, e.g., a mobile X-ray device (C-arm). Different registration approaches can be used depending on the used modalities, the particular surgical procedure, and organ properties to be registered.

This work discusses an intensity-based 2D/3D registration approach using pre-operative CT data and 2D X-ray images for minimally invasive spine surgery. The intensity-based 2D/3D registration procedure is mathematically characterized and decomposed in its essential elements. A cost function is created using the comparison of X-ray images and digitally reconstructed radiographs (DRR) created out of the CT data. An optimization algorithm is used to minimize the cost function and find the registration pose. The DRR rendering is found to be computationally expensive, being the registration bottleneck. A novel optimization based on parallel computing is applied to the DRR process.

A set of reference frames supported by the navigation system is considered to transform the registration results into the OR. The C-arm reference frame is described using the pinhole camera model and found with a parametrization device. The parametrization device is developed based on a simulation analysis of accuracy vs. size and evaluated based on an innovative inverse registration approach.

Additionally, two novel improvements to the registration procedure are made using deep-learning. X-ray images captured using image intensifier C-arms are prone to distortions due to the earth's magnetic field. The implemented undistortion uses an image warping process that requires a plate with steel fiducials installed on the C-arm image detector. Each fiducial in the resulted X-ray images is detected with a convolutional neural network (CNN). Previously to the 2D/3D registration execution, it is required to input an initial pose. A graphical approach for the manual selection of the initial pose is implemented, but also an automatic initial pose generator based on a CNN is developed.

The combination of implemented procedures is a fully automated local 2D/3D registration with an average accuracy of 1.5 mm, measured with the navigation system. The result of the registrations can be transferred easily to a navigated robot system.

Zusammenfassung

Die minimalinvasive Chirurgie durch Fortschritte in der medizinischen Bildgebung ermöglicht, die im Vergleich zur konventionellen offenen Chirurgie zu einer schnelleren Durchführung des Eingriffs, geringeren Schnittgrößen und Knochenfreilegung sowie schnelleren Erholungszeiten führt. Ein besonders komplexer minimalinvasiver Eingriff ist das Einsetzen von Pedikelschrauben in der Lendenwirbelsäule, der eine Genauigkeit der Klasse A oder B auf der Gertzbein-Robbins-Skala erfordert, d.h. unter 2 mm. Ein minimalinvasiver chirurgischer Eingriff beginnt mit der präoperativen Bildgebung, z. B. einer 3D-Computertomographie (CT), die in der Regel außerhalb des Operationssaals (OP) aufgenommen und zur Diagnose und Planung der Posen für die einzusetzenden Pedikelschrauben verwendet wird. Eine große Herausforderung besteht darin, die geplanten Daten in den OP zu bringen, d. h. die präoperativen Daten mit der aktuellen Patientenposition zu registrieren, wobei allgegenwärtige Bildgebungsmodalitäten verwendet werden, die in Operationssälen zu finden sind, wie beispielsweise ein mobiles Röntgengerät (C-Bogen). Abhängig von den verwendeten Modalitäten, dem jeweiligen chirurgischen Verfahren und den zu registrierenden Organeigenschaften können unterschiedliche Registrierungsansätze verwendet werden.

In dieser Arbeit wird ein intensitätsbasierter 2D/3D-Registrierungsansatz unter Verwendung von präoperativen CT-Daten und 2D-Röntgenbildern für die minimalinvasive Wirbelsäulenchirurgie diskutiert. Das intensitätsbasierte 2D/3D-Registrierungsverfahren wird mathematisch charakterisiert und in seine wesentlichen Elemente zerlegt. Aus dem Vergleich von Röntgenbildern und digital rekonstruierten Röntgenbildern (DRR), die aus den CT-Daten erstellt werden, wird eine Kostenfunktion erstellt. Ein Optimierungsalgorithmus wird verwendet, um die Kostenfunktion zu minimieren und die Registrierungspose zu finden. Es hat sich herausgestellt, dass das DRR-Rendering rechenintensiv ist und den Engpass bei der Registrierung darstellt. Eine neuartige Optimierung, die auf parallelem Rechnen basiert, wird auf den DRR-Prozess angewendet.

Ein Satz von Koordinatensystemen, die vom Navigationssystem unterstützt werden, wird berücksichtigt, um die Registrierungsergebnisse in den OP zu transformieren. Das C-Bogen-Koordinatensystem wird mit Hilfe des Lochkameramodells beschrieben und mit einem Parametrisierungsvorrichtung gefunden. Die Parametrisierungsvorrichtung wird auf der Grundlage einer Simulationsanalyse von Genauigkeit vs. Größe entwickelt und auf der Basis eines innovativen inversen Registrierungsansatzes bewertet.

Zusätzlich werden zwei neuartige Verbesserungen des Registrierungsverfahrens mittels Deep-Learning vorgenommen. Röntgenbilder, die mit Bildverstärker-C-Bögen aufgenommen werden, sind anfällig für Verzerrungen aufgrund des Erdmagnetfeldes. Die Bilder werden durch eine Bildentzerrung korrigiert, die eine Platte mit Stahl-Markern auf dem C-Bogen benötigt. Jeder Marker in den resultierenden Röntgenbildern wird mit einem Convolutional Neural Network (CNN) erkannt. Vor der Ausführung der 2D/3D-Registrierung muss eine Ausgangsposition eingegeben werden. Es wird ein grafischer Ansatz für die manuelle Auswahl der Anfangs-Pose implementiert, aber auch ein automatischer Generator für die Anfangs-Pose auf Basis eines CNN entwickelt.

Die Kombination der implementierten Verfahren ist eine vollautomatische lokale 2D/3D-Registrierung mit einer durchschnittlichen Genauigkeit von 1,5 mm, gemessen mit dem Navigationssystem. Das Ergebnis der Registrierungen kann mit geringem Aufwand auf ein navigiertes Robotersystem übertragen werden.

Abbreviations

AD	AdaDelta
Adam	Adaptive Moment Estimation
AG	AdaGrad
AP	Anterior-Posterior
ARB	Aim Rigid Body (RB attached to the patient)
BN	Best Neighbors
CHT	Circle Hough Transform
CNN	Convolutional Neural Network
CR	Correlation Ratio
CRB	Calibration Rigid Body (RB attached to the C-arm parametrization device)
CT-scan	Computed Tomography Scan
CUDA	Compute Unified Device Architecture
DICOM	Digital Imaging and Communications in Medicine
DoF	Degrees of freedom
DRB	Device Rigid Body (RB attached to the 2D/3D registration testing device)
DRR	Digitally Reconstructed Radiograph
FCNN	Fully Connected Neural Network
GC	Gradient Correlation
GD	Gradient Difference
GRDE	Gradient Descent
LAT	Lateral
MI	Mutual Information
MR	Magnetic Resonance
NACC	Normalized Absolute Cross-Correlation
NCC	Normalized Cross-Correlation
NMI	Normalized Mutual Information
OR	Operating Room
OTS	Optical Tracking System (reference frame of the stereo system)
PI	Pattern Intensity
PRB	Pointer Rigid Body
RB	Rigid Body
ReLU	Rectified Linear Unit
SAD	Sum of Absolute Differences
SLNCC	Sum of Local Normalized Cross-Correlation

SSD	Sum of Squared Differences
TRB	Tool Rigid Body (RB attached to the robot end-effector)
VWSLNC	Variance-Weighted Sum of Local Normalized Correlation

Mathematical Notation

$\mathbb{C}$	The Set of Complex Numbers
$\mathbb{H}$	Quaternion Space
$\wp^{\boldsymbol{n}}$	Real Projective n-space
$\mathbb{R}$	The Set of Real Numbers
$\mathbb{R}^{\boldsymbol{n}}$	Euclidean n-space
$\mathbb{Z}$	The Set of Integer Numbers
$\mathbf{A}$	Matrix $\mathbf{A}$
$\vec{\boldsymbol{b}}$	Column Vector $\vec{\boldsymbol{b}}$
$\boldsymbol{c}$	Point $\boldsymbol{c} \in \mathbb{R}^n$
$\tilde{\boldsymbol{c}}$	Point $\tilde{\boldsymbol{c}} \in \wp^n$
d	scalar d
$\mathbf{q}$	Quaternion $\mathbf{q}$
$\mathbf{T} = \{a_0, \dots, a_n\}$	Set $\mathbf{T}$ of $n+1$ elements. It contains elements from a_0 to a_n
{A}	Reference frame A
${}^{A}\boldsymbol{p}_B$	Coordinate of {B} with respect to {A}
${}^{A}\boldsymbol{R}_B$	Rotation matrix from {B} with respect to {A}
${}^{A}\boldsymbol{T}_B$	Transformation matrix from {B} with respect to {A}

List of Figures

1. Introduction

Insertion of pedicle screws is a medical procedure that requires cutting a significant segment of the patient's body to expose the bone structure in conventional open surgery. With the full sight of the bone segment, surgeons can proceed to insert pedicle screws. This procedure creates a trauma that reflects in several months of post-operative recovery. Advances in medical imaging technologies motivated the creation of *minimally invasive surgery*, which leads to a faster procedure execution, reduced incision size and bone exposure, and less traumatic and quicker recovery.

A minimally invasive surgical procedure starts with pre-operative imaging acquisition, e.g., a 3D-computed tomography (CT-scan), usually taken outside of the operating room. Using the planning software, the patient is diagnosed, and the insertion of pedicle screws is planned. At the beginning of the surgical procedure, the registration stage is crucial to bringing pre-operative data into the intra-operative scenario. Conventional matching procedures like paired-point or surface matching procedures are unsuitable in the spine application because it is hardly possible to identify landmarks or surfaces accessible during the intervention. Image registration approaches do not have this drawback, but they require an imaging modality in the operating room. In a typical *2D/3D registration*, intra-operative 2D imaging are acquired using a ubiquitous modality available in every operating room, a mobile X-ray unit, also known as C-arm. The 2D/3D registration result links the pre-operative data with the patient's position, whose reference frame is tracked with a navigation system. The screw insertion can be accomplished using pre-operative planned data together with the surgical robot system.

In this work, an intensity-based *2D/3D registration* procedure for lumbar spine surgery is implemented. This work focuses on evaluating and optimizing the typical elements of the registration procedure that maximize the pedicle screw accuracy in minimally invasive lumbar spine surgery. The evaluation criteria used during the implementation process lead to a unique 2D/3D registration approach. The implemented optimizations enable the registration to be carried out in few seconds. Among these optimizations, novel approaches for automating the registration procedure are developed.

The implementation of the 2D/3D registration can be seen as part of two main components. The first component consists of the C-arm characterization, i.e., applying a C-arm mathematical model that links an X-ray image with the volume position in the 3D space. This characterization also finds the transformation of the C-arm coordinate system with a known reference frame. In the second component, the registration procedure must ensure a match between the 3D modality and the X-ray images. The result given for this stage provides the connection between the 3D and 2D modalities, i.e., the pre- and intra-operative modalities. With the knowledge of the C-arm coordinate frame, the previous result is transformed into a coordinate frame system inside the operating theater.

The first part of this work elaborates on the C-arm mathematical model, the reference frames involved in the 2D/3D registration, and image undistortion from C-arms with image intensifier technology. The second part explains the 2D/3D registration process and the development of fundamental components such as image similarity measurements, optimization algorithms, and the digital reconstructed radiograph module. At the end of the second part, an experimental chapter shows the 2D/3D registration accuracy. The last section of this work discusses the improvements in the image undistortion process and the automatic selection of the initial pose for the 2D/3D registration process using deep-learning techniques.

2. Motivation to Evaluate and Optimize Registration Procedures for Minimally Invasive Spine Surgery with Navigated Surgical Robot Systems

The incorporation of new technologies into the medical field has led some surgical interventions to be implemented as minimally invasive procedures. One of the first known minimally invasive procedures is the endoscopy, which allows the analysis of the body's interior with a small incision, e.g., laparoscopy, or even without any cut, e.g., colonoscopy. During the last decades, the incorporation of optical stereo vision systems (navigation systems) and surgical robots have increased the potential of minimally invasive surgery to insert pedicle screws in the spine accurately. The requirements that have to be fulfilled by the developed procedures to be incorporated into robotic systems are discussed at the end of this chapter. The consequences of using minimally invasive surgery and the registration process that is done, previously to the execution of the navigated surgical procedure, are explained in the following section. In the end, the task consists of developing a transformation matrix that allows converting a planned pose in the pre-operative data, e.g., a pedicle screw pose, into a target pose for the navigated robot system.

2.1. Considerations of Minimally Invasive Surgery with Navigated Surgical Robot Systems

The use of minimally invasive surgery brings some advantages in comparison with open surgery interventions. From the patient's perspective, pain, hospital stay, and complication reduction are among the advantages of minimally invasive surgeries compared with open surgeries [1]. Surgeons also benefit from minimally invasive procedures as the amount of radiation they received is reduced [2].

On the other hand, some drawbacks arise from the inclusion of minimally invasive surgery; for example, the acquisition cost of a minimally invasive setup for an operating room is high [3]. Apart from that, some technical challenges from the implementation stand of view have to be considered for developing a minimally invasive surgery. They are split into image registration challenges and target requirements for the navigated surgical robot system. This division is considered because the planning software, which includes the registration process, and the navigated surgical robot are the two macro components of a minimally invasive surgery with navigated surgical robot systems.

2.2. Technical Challenges in Image Registration: Implementation and Optimization

As the required modalities and the patient set up in a minimally invasive surgery is achieved once the medical procedure is already started, i.e., the patient is anesthetized, and some incisions are already done [4], the execution time of the registration is critical. That means the registration procedure has to be carried out as fast as possible. Depending on the taken approach, some registrations use iterative methodologies, which are computationally expensive. Some approaches report registration times of some minutes [5]. Though using modern hardware, such as top-notch CPUs, helps decrease the execution time of the registration, another paradigm like parallel computing, using GPUs, presents a better chance of reducing times to just a couple of seconds [6].

In the planning stage of the minimally invasive procedure, a medical modality, e.g., CT-scan, taken days before the surgery (pre-operative modality) can be used to define the locations of the implants [1]. There is no life risk for the patient at this stage, so the interaction with the planning software can be done without significant concern. The main challenge of the registration consists of bringing the planned data into the operating scenario using modalities inherent to the operating theater (intra-operative modality), e.g., mobile X-ray device.

A technical challenge derived from the above paragraph can be described as minimizing the human intervention in importing pre-operative data to the operating scenario using intra-operative modalities. In other words, the autonomy of the registration procedure has to be as high as possible while assuring the registration accuracy.

In open surgeries, where a navigation system is used for feedbacking the position and orientation of instruments (optically navigated surgery), surgeons can verify the accuracy of registered modalities using distinguishable features of exposed bones [7]. This accuracy assessment cannot be done in minimally invasive surgery. The quality of a registration cannot be evaluated per se, requiring external and indirect measurements to validate, e.g., start the medical procedure and use X-ray images to verify that the trajectories of the perforations go according to the plan. The implementation of a registration process has to be tested, such as it guarantees an accuracy below the expected tolerance, less than 2mm for spine implant positioning [1].

The implementation of the registration process for minimally invasive spine surgery introduces additional elements into the operating theater, e.g., optical tracking system, rigid bodies (RB) attached to the patient, calibration devices, and others. It is desired that the number of these elements and their interference with the surgical workflow is minimal. The introduction of new devices induces additional procedures on top of the standard medical protocols. It is necessary to develop a registration process that limits the number of new elements and adds as few additional steps as possible to the surgical workflow.

2.3. Target Requirements for Navigated Surgical Robot Systems

This work is framed on the SIISpine project, which focuses on developing a robotic assistance system to support minimal invasive spine interventions with reduced radiation. The SIISpine project development is supported by the previously developed modular interactive computer-assisted surgery (modiCAS) project. Some contributions to the modiCAS project have been focused on developing a solution for an integrated system consisting of a navigation component and a robotic component, resulting in a so-called "navigated robot system." This navigated surgical robot system receives inputs like a conventional navigation system without a robot. The robot arm can be considered as a mechatronic extension of the navigation system. Due to the inherent characteristics of the modiCAS system, the robot arm is positioned in the pose sent to the navigation component. Using standard industrial robot arms, the achievable resolution and accuracy of this integral system are similar to the values achievable for the navigation component if used without a robot arm, i.e., they mainly depend on the accuracy of the used stereo camera and DRB design [8]. It means the registration results can be used with the pre-operative data and the navigation component to supply the robot system aim pose.

The registration accuracy is found in this thesis using the navigation system without involving the robot system, i.e., to avoid the risk that the robot introduces some errors. In conclusion, although a surgical robot later uses the registration results, this work centers its attention on computing the transformation from pre-operative data into an aim pose for the navigated robot system and guaranteeing accurate registrations that leads to implant placement with grades A or B on the Gertzbein-Robbins scale, i.e., below 2mm [9].

3. Scope and Objectives of the Evaluation and Optimization of 2D/3D Registration

This thesis focuses on implementing a fully automatic intensity-based 2D/3D registration for lumbar spine using pre-operative CT-scans and intra-operative X-ray modalities. Some technical restrictions are introduced, such as limiting additional elements inside the operating theater, e.g., rigid bodies, fiducials on the patient, helping structures. These restrictions do not exclude the registration accuracy for lumbar spine surgery that guarantees implants with grades A or B on the Gertzbein Robbins scale. The main scope of this work can be systematically split into the following objectives:

- Finding the mathematical procedure to perform the 2D/3D registration and transforming the results into a useful frame for a surgical robot system: This objective is developed with the combined outcome of sections 5.1 and 5.2. In section 5.1, reference frames involved in the 2D/3D are introduced, and the equation of the 2D/3D registration is deduced. The structure of the 2D/3D registration procedure is presented in section 5.2. The analysis of including a region of interest during the registration is developed in section 5.8.
- Characterizing the intra-operative imaging device, i.e., mobile X-ray device (C-arm), and correcting and processing the obtained intra-operative images: X-ray images of image intensifier C-arms present distortions. The required elements to find the distortions and the undistortion approach are presented in section 5.3. The C-arm is characterized using the pinhole camera model; the implications, requirements, and results for this characterization are described in section 5.4.
- Developing and selecting the fundamental components of the intensity-based 2D/3D registration process, e.g., cost function, optimization algorithms, and digitally reconstructed radiograph (DRR): The evaluation of cost functions and optimization algorithms for the 2D/3D registration is carried out in section 5.5, and an available DRR is evaluated in the first part of section 5.6. A graphical interface to select the initial seed for the optimization is developed in section 5.7.
- Optimizing bottle-neck processes that restrict the registration accuracy and execution time: An improved and faster DRR using parallel computing is implemented in the last part of section 5.6. The X-ray imaging undistortion process is improved using deep-learning in section 7.1. An automatic initial pose generator for the 2D/3D registration is developed in section 7.2.

4. State of Art of Image Registration Approaches and Imaging Modalities

4.1. Minimally Invasive Surgery

The concept of minimally invasive surgery is adopted for medical interventions with reduced body damage employing techniques and technology that avoid open surgeries. Minimally invasive surgery offers several advantages for the patients, including less pain, less blood loss, shorter hospital stays, faster recovery, and fewer complications [1].

One of the very first developed minimally invasive surgery was the laparoscopy, which is done through a set of small incisions, using micro-cameras and tiny surgical instruments. This kind of technology still requires a specialist to operate the tools. Consequently, it is classified as non-robotic minimally invasive surgery.

In the case of pedicle screw insertion in the spine, non-robotic minimally invasive surgery uses Kirscher wires (K-wires) and a continuous X-ray image, known as fluoroscopy, to guide the insertion [10]. Although this procedure has shown accuracies comparable to those of open spine surgery, it exposes the surgeons to high doses of radiation. When a robot can assist the insertion, the process is then classified as robotic minimally invasive surgery, bringing the additional advantage of no-radiation for the surgeon and fine movement of the end-tools [2].

Within robotic minimally invasive surgery, there are master-slave robots, and computer-assisted navigated robots. Minimally invasive surgery with a master-slave robot is the case when the physician entirely manipulates the robot. The most known system of this kind is the da Vinci surgical robotic system. This system improves the visual field, increases the surgeon's movement precision, and boosts the physician endurance in long surgeries [4]. This type of system has one general pitfall: the cost of the solution [3]. These systems also lack robot autonomy since the system behaves like a manipulator. In some applications like spine surgery, the robotic system is expected to play a more active role. These drawbacks are overcome by the computer-assisted navigated robots, which is the topic of section 4.3.

4.2. Position Vectors, Rotation and Homogenous Transformations Matrices

Before going any further with the concepts involved in the development of this work, it is essential to introduce the notations used in this work for position vectors, rotation matrices, quaternions, and affine transformations.

4.2.1. Position Vectors and Rotation Matrices

Let {A} be a reference frame, and $\boldsymbol{c} \in \mathbb{R}^3$ be a point. $\boldsymbol{c}$ is described in {A} with the notation,

$$ {}^{A}\boldsymbol{p}_c = \begin{bmatrix} {}^{A}x_c \\ {}^{A}y_c \\ {}^{A}z_c \end{bmatrix} \tag{1}$$

which expresses c as the position vector with cartesian coordinates $[x, y, z]^T$ with respect to {A}. Figure 4-1 illustrates the previous concept.

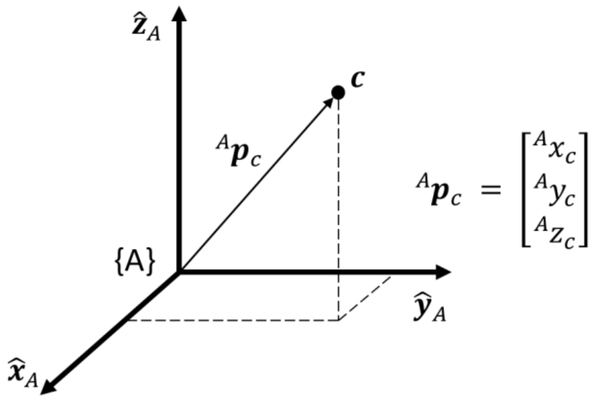

Figure 4-1. Illustration of the position vector ${}^{A}\boldsymbol{P}_c$

Let be {A} and {B} be two independent reference frames. The position of the frame {B} with respect to {A} is given by ${}^{A}\boldsymbol{p}_B$. The reference frame {B} is defined by three orthonormal vectors $\hat{\boldsymbol{x}}_B, \hat{\boldsymbol{y}}_B, \hat{\boldsymbol{z}}_B$, which are known in {A}, i.e., ${}^{A}\hat{\boldsymbol{x}}_B, {}^{A}\hat{\boldsymbol{y}}_B, {}^{A}\hat{\boldsymbol{z}}_B$. See Figure 4-2.

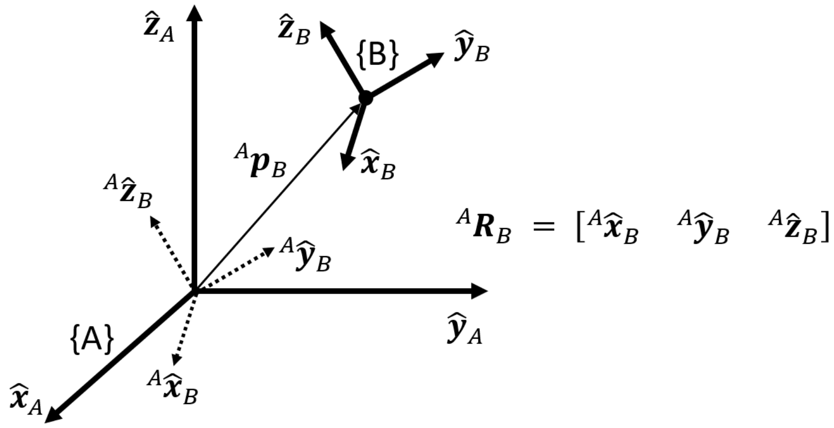

Figure 4-2. Illustration of the rotation ${}^{A}\boldsymbol{R}_B$

There three orthonormal vectors are used as the columns of a 3x3 matrix:

$$ {}^{A}\boldsymbol{R}_B = [{}^{A}\hat{\boldsymbol{x}}_B \quad {}^{A}\hat{\boldsymbol{y}}_B \quad {}^{A}\hat{\boldsymbol{z}}_B] \tag{2}$$

The matrix ${}^{A}\boldsymbol{R}_B$ is known as a rotation matrix, which has the following properties:

- A rotation matrix is an orthonormal matrix. It means each column vector is orthogonal with the others, the magnitude of each column vector is 1, and the rotation matrix determinant is 1.

- The rows of a rotation matrix, e.g., ${}^{A}\boldsymbol{R}_{B}$, give the three orthonormal vectors of {A} relative to {B}, i.e., ${}^{B}\hat{\boldsymbol{x}}_{A}$, ${}^{B}\hat{\boldsymbol{y}}_{A}$, ${}^{B}\hat{\boldsymbol{z}}_{A}$. Therefore, the rotation of {A} related to {B} is found by transposing rows and columns, that is ${}^{B}\boldsymbol{R}_{A}$ [11]. In other words, the inverse of a rotation matrix is equal to its transposed one:

$$ {}^{B}\boldsymbol{R}_{A} = {}^{A}\boldsymbol{R}_{B}^{-1} = {}^{A}\boldsymbol{R}_{B}^{T} = \begin{bmatrix} {}^{A}\hat{\boldsymbol{x}}_{B}^{T} \\ {}^{A}\hat{\boldsymbol{y}}_{B}^{T} \\ {}^{A}\hat{\boldsymbol{z}}_{B}^{T} \end{bmatrix} \tag{3} $$

A set of fundamental rotation matrices describe the rotations around X-, Y- and Z-axis by a given angle. These matrices are known as basic rotation matrices and are defined as follows:

$$ Rot\{x,\alpha\} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & cos\alpha & -sin\alpha \\ 0 & sin\alpha & cos\alpha \end{bmatrix} \tag{4} $$

$$ Rot\{y,\beta\} = \begin{bmatrix} cos\beta & 0 & sin\beta \\ 0 & 1 & 0 \\ -sin\beta & 0 & cos\beta \end{bmatrix} \tag{5} $$

$$ Rot\{z,\gamma\} = \begin{bmatrix} cos\gamma & -sin\gamma & 0 \\ sin\gamma & cos\gamma & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{6} $$

These equations produce right-hand rotations. In total, there are twelve possible combinations with those angles. As there are three rotational DoF in the 3D-space, a 3-DoF-rotation can be expressed as the product of three consecutive basic rotation matrices. Let the coordinate frame {A} be rotated around the Z-axis by an angle γ, $Rot\{z,\gamma\}$, then in the resulting X'-axis by an angle α, $Rot\{x',\alpha\}$, and a final rotation around the resulting Y''-axis by an angle β, $Rot\{y'',\beta\}$. This chain of rotation can be seen in Figure 4-3.

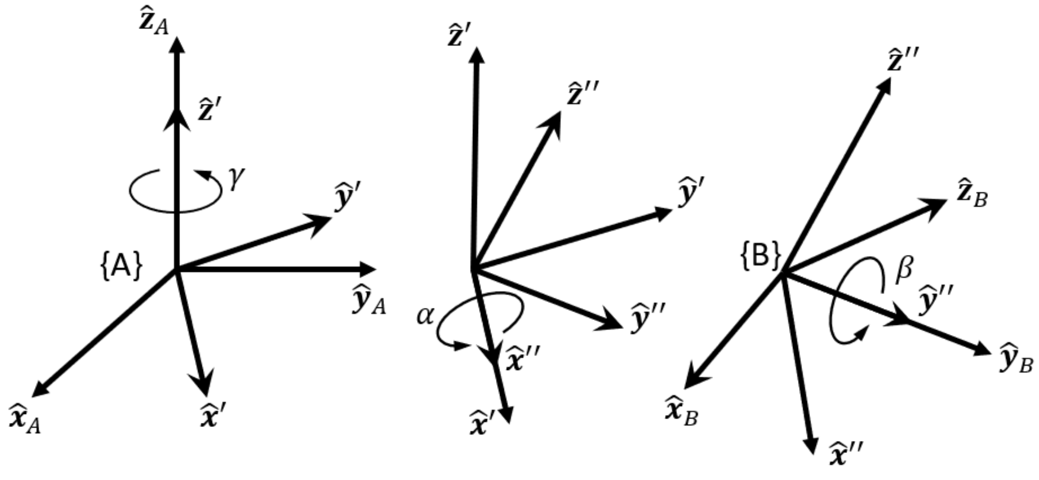

Figure 4-3. $Z - X' - Y''$ rotation

This chain of rotations is known as the $Z - X' - Y''$ representation. In this case, the resulted reference frame is used for the next rotation, known as intrinsic rotations. There is another approach called extrinsic rotations, but it will not be discussed as they are not used to develop this work. The $Z - X' - Y''$ rotation can be expressed mathematically as in (7).

$$
\begin{aligned}
{}^{A}\boldsymbol{R}_{B,ZXY} &= \mathrm{Rot}\{z,\gamma\}\cdot\mathrm{Rot}\{x',\alpha\}\cdot\ \mathrm{Rot}\{y'',\beta\} \\
&= \begin{bmatrix} cos\gamma\cdot cos\beta - sin\gamma\cdot sin\alpha\cdot sin\beta & -sin\gamma\cdot cos\alpha & cos\gamma\cdot sin\beta + sin\gamma\cdot sin\alpha\cdot cos\beta \\ sin\gamma\cdot cos\beta + cos\gamma\cdot sin\alpha\cdot sin\beta & cos\gamma\cdot cos\alpha & sin\gamma\cdot sin\beta - cos\gamma\cdot sin\alpha\cdot cos\beta \\ -cos\alpha\cdot sin\beta & sin\alpha & cos\alpha\cdot cos\beta \end{bmatrix}
\end{aligned} \tag{7}
$$

For sake of clarification, the rotation matrix in *(7)* is called ${}^{A}\boldsymbol{R}_{B,ZXY}$. Later on, a rotation matrix, e.g., from frame {A} to {B}, will be named ${}^{A}\boldsymbol{R}_{B}$ regardless the used representation for the rotation.

4.2.2. 3D Affine Transformation Matrices

Now let the relation between {A} and {B} be known, i.e., translation, ${}^{A}\boldsymbol{p}_{B}$, and orientation, ${}^{A}\boldsymbol{R}_{B}$. In this case, the point $\boldsymbol{c}$ is described in {B}, that is ${}^{B}\boldsymbol{p}_{c}$. The before mentioned elements are shown in Figure 4-4.

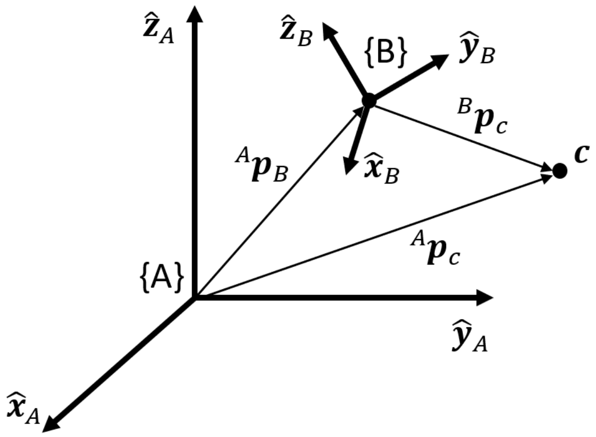

Figure 4-4. Transformation of the point *c* in {B}, ${}^{B}\boldsymbol{P}_{c}$, to {A}, ${}^{A}\boldsymbol{P}_{c}$

Mathematically, ${}^{A}\boldsymbol{p}_{c}$ can be found by translating {A} by ${}^{A}\boldsymbol{p}_{B}$, then rotating {A} until it aligns with {B} using ${}^{A}\boldsymbol{R}_{B}$. Finally, the point $\boldsymbol{c}$ is found by knowing its position in {B}, ${}^{B}\boldsymbol{p}_{c}$. See equation *(8)*.

$$ {}^{A}\boldsymbol{p}_{c} = {}^{A}\boldsymbol{p}_{B} + {}^{A}\boldsymbol{R}_{B}\cdot {}^{B}\boldsymbol{p}_{c} \tag{8} $$

From *(8)*, the transformation is simplified as,

$$ {}^{A}\boldsymbol{p}_{c} = {}^{A}\boldsymbol{T}_{B}\cdot {}^{B}\boldsymbol{p}_{C} \tag{9} $$

where ${}^{A}\boldsymbol{T}_{B}$ is built out of ${}^{A}\boldsymbol{p}_{B}$ and ${}^{A}\boldsymbol{R}_{B}$. The matrix ${}^{A}\boldsymbol{T}_{B}$ is known as an affine transformation matrix, and it is also named as the pose of B (with respect to A). The affine transformation matrix, also referred to as the transformation matrix, is a 4x4 matrix. The last row of the transformation matrix is a constant $[0\ 0\ 0\ 1]$ row vector. The structure of ${}^{A}\boldsymbol{T}_{B}$ can be seen in equation *(10)*.

$$ \begin{bmatrix} {}^{A}\boldsymbol{p}_{c} \\ 1 \end{bmatrix} = \begin{bmatrix} & {}^{A}\boldsymbol{R}_{B} & & {}^{A}\boldsymbol{p}_{B} \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} {}^{B}\boldsymbol{p}_{c} \\ 1 \end{bmatrix} \tag{10} $$

In case that ${}^{B}\boldsymbol{p}_{c}$ is the position to be calculated using equation *(8)*, it can be found that:

$$ {}^{B}\boldsymbol{p}_{c} = -{}^{B}\boldsymbol{R}_{A}\cdot {}^{A}\boldsymbol{p}_{B} + {}^{B}\boldsymbol{R}_{A}\cdot {}^{A}\boldsymbol{p}_{c} \tag{11} $$

From *(11)*, the transformation is simplified as:

$$ {}^{B}\boldsymbol{p}_c = {}^{B}\boldsymbol{T}_A \cdot {}^{A}\boldsymbol{p}_c \tag{12}$$

The matrix ${}^{B}\boldsymbol{T}_A$ can be made as equation *(10)*

$$\begin{bmatrix} {}^{B}\boldsymbol{p}_c \\ 1 \end{bmatrix} = \begin{bmatrix} & {}^{B}\boldsymbol{R}_A & & -{}^{B}\boldsymbol{R}_A \cdot {}^{A}\boldsymbol{p}_B \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} {}^{A}\boldsymbol{p}_c \\ 1 \end{bmatrix} \tag{13}$$

From *(9)* and *(12)*, it can be said that

$$ {}^{A}\boldsymbol{T}_B = {}^{B}\boldsymbol{T}_A^{-1} \tag{14}$$

and from *(10)* and *(13)*, it can be found that [11]

$$ {}^{B}\boldsymbol{T}_A^{-1} = \begin{bmatrix} & {}^{A}\boldsymbol{R}_B^{-1} & & {}^{B}\boldsymbol{p}_A \\ 0 & 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} & {}^{B}\boldsymbol{R}_A & & -{}^{B}\boldsymbol{R}_A \cdot {}^{A}\boldsymbol{p}_B \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{15}$$

4.2.3. Chain of Affine transformation matrices

Let {A}, {B},…, {M}, {N} be a finite number of reference frames as depicted in Figure 4-5. The transformations ${}^{A}\boldsymbol{T}_B$,…, ${}^{M}\boldsymbol{T}_N$ and the point $\boldsymbol{q}$ in {N} are known, ${}^{N}\boldsymbol{p}_q$.

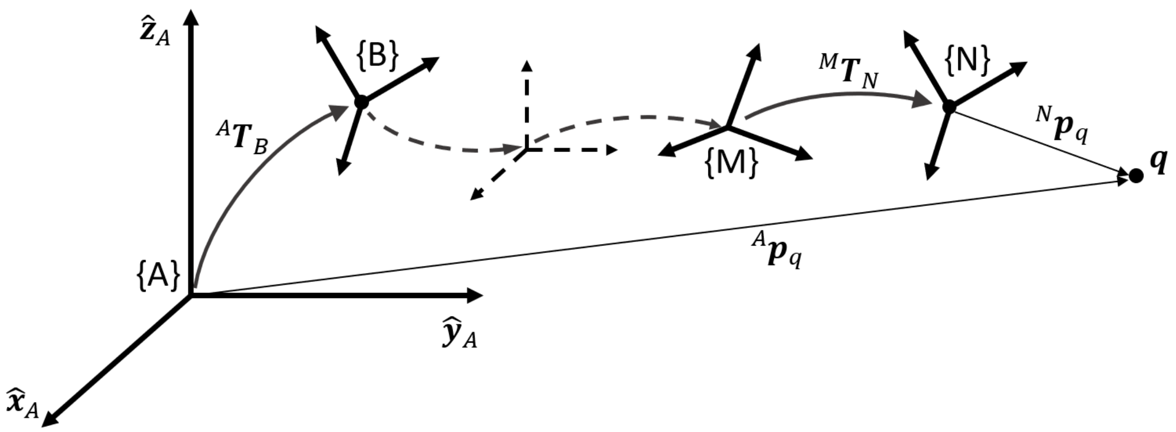

Figure 4-5. Transformation using a finite number of reference frames

The point $\boldsymbol{q}$ is desired to be known in {A}, ${}^{A}\boldsymbol{p}_q$. The mathematical development is done in equation *(16)*

$$ {}^{A}\boldsymbol{p}_q = {}^{A}\boldsymbol{T}_B \cdots {}^{M}\boldsymbol{T}_N \cdot {}^{N}\boldsymbol{p}_q \tag{16}$$

equation *(16)* can be further simplify as:

$$ {}^{A}\boldsymbol{p}_q = {}^{A}\boldsymbol{T}_N \cdot {}^{N}\boldsymbol{p}_q \tag{17}$$

Where ${}^{A}\boldsymbol{T}_{\mathrm{N}}$ is the product of the transformations $\prod_{i=A}^{N} {}^{i}\boldsymbol{T}_{i+1}$.

An important conclusion from *(10)* and *(17)* indicates that ${}^{A}\boldsymbol{p}_N$ and ${}^{A}\boldsymbol{R}_N$ are also known once ${}^{A}\boldsymbol{T}_N$ is calculated. This result is used in several sections in chapter 5 when a transformation matrix is calculated, e.g., ${}^{A}\boldsymbol{T}_{N}$, the position and the rotation from the initial to the final frame, e.g., ${}^{A}\boldsymbol{p}_{N}$ and ${}^{A}\boldsymbol{R}_{N}$, are used in the following steps without further clarification.

4.2.4. Quaternions

Rotations in 3D using quaternions follow a different approach than Euler angles rotations. Instead of having a set of three subsequent intrinsic rotations, rotations with quaternions are done only by one 3D rotation. This single rotation uses one vector representing the rotation axis and one scalar as the rotation angle. The following mathematical formalisms are introduced to describe quaternions and their basic operations:

Let $\mathbb{H}$ be the quaternion space, which is defined as:

$$\mathbb{H} = \{a + b\hat{\boldsymbol{\imath}} + c\hat{\boldsymbol{\jmath}} + d\widehat{\boldsymbol{k}}\ ;\ a, b, c, d\ \in \mathbb{R}\ |\ \hat{\boldsymbol{\imath}}^2 = \hat{\boldsymbol{\jmath}}^2 = \widehat{\boldsymbol{k}}^2 = ijk = -1\} \tag{18}$$

Let $\mathbf{q} \epsilon \mathbb{H}$ be a quaternion defined as

$$\mathbf{q} = s + v_x\hat{\boldsymbol{\imath}} + v_y\hat{\boldsymbol{\jmath}} + v_z\widehat{\boldsymbol{k}} \tag{19}$$

$\mathbf{q}$, and generally, a quaternion, can be further grouped as a scalar and a vector

$$\mathbf{q} = (s, \vec{\boldsymbol{v}}) \tag{20}$$

where $\vec{\boldsymbol{v}} = v_x\hat{\boldsymbol{\imath}} + v_y\hat{\boldsymbol{\jmath}} + v_z\widehat{\boldsymbol{k}}$

The following relations are valid for the calculations with quaternions [11]:

a) Let $\mathbf{q}_1, \mathbf{q}_2, \mathbf{q}_3 \epsilon \mathbb{H}$. The product of two quaternions results in a quaternion, i.e., $\mathbf{q}_1 \cdot \mathbf{q}_2 = \mathbf{q}_3$. The product is associative but not commutative, and it is defined as:

$$\mathbf{q}_3 = \mathbf{q}_1 \cdot \mathbf{q}_2 = (s_1 s_2 - \vec{\boldsymbol{v}}_1 \cdot \vec{\boldsymbol{v}}_2,\ s_1\vec{\boldsymbol{v}}_2\ +\ s_2\vec{\boldsymbol{v}}_1 + \vec{\boldsymbol{v}}_1 \times \vec{\boldsymbol{v}}_2) \tag{21}$$

b) Let $\mathbf{q}, \mathbf{e}\ \epsilon \mathbb{H}$ such as:

$$\mathbf{q} \cdot \mathbf{e} = \mathbf{e} \cdot \mathbf{q} = \mathbf{q}, \quad \mathbf{e}\ =\ (1, \mathbf{0}) \tag{22}$$

c) Let $\mathbf{q} \epsilon \mathbb{H}$, then it exists an inverse quaternion $\mathbf{q}^{-1}$, such as:

$$\mathbf{q} \cdot \mathbf{q}^{-1} = \mathbf{q}^{-1} \cdot \mathbf{q} = \mathbf{e},$$
$$\mathbf{q}^{-1} = \frac{\mathbf{1}}{s^2 + \vec{\boldsymbol{v}}^2}(s, -\vec{\boldsymbol{v}}) \tag{23}$$

where $\vec{\boldsymbol{v}}^2 = {v_x}^2 + {v_y}^2 + {v_z}^2$

d) Let $\mathbf{q} \epsilon \mathbb{H}$ be a unit quaternion, which has the following property:

$$s^2 + \vec{\boldsymbol{v}}^2 = 1 \tag{24}$$

In a unit quaternion, the scalar, s, describes the rotational angle, and the vector, $\vec{\boldsymbol{v}}$, describes the rotational normal axis $\hat{\boldsymbol{n}}$.

$\mathbf{q} = (s, \vec{\boldsymbol{v}}) = (s, v_x, v_y, v_z)$ can be written as:

$$\mathbf{q} = \left[cos\left(\frac{\theta}{2}\right), sin\left(\frac{\theta}{2}\right)\hat{\boldsymbol{n}}\right] = \left[cos\left(\frac{\theta}{2}\right), sin\left(\frac{\theta}{2}\right)n_x, sin\left(\frac{\theta}{2}\right)n_y, sin\left(\frac{\theta}{2}\right)n_z\right] \tag{25}$$

where $cos\left(\frac{\theta}{2}\right) = s,\ sin\left(\frac{\theta}{2}\right)\hat{\boldsymbol{n}} = \vec{\boldsymbol{v}}$

Let $\vec{\boldsymbol{u}}, \vec{\boldsymbol{u}}' \epsilon \mathbb{R}^3$ be vectors, $\hat{\boldsymbol{n}} \epsilon \mathbb{R}^3$ be a normal vector and $\theta \epsilon \mathbb{R}$ the angle around $\hat{\boldsymbol{n}}$ where $\vec{\boldsymbol{u}}$ will be rotated. The rotated vector $\vec{\boldsymbol{u}}$ is expressed as $\vec{\boldsymbol{u}}'$. The rotation can be seen in Figure 4-6 , and is mathematically expressed in equation (*26*).

$$\vec{\boldsymbol{u}}' = \mathbf{q} \cdot \vec{\boldsymbol{u}} \cdot \mathbf{q}^{-1} = \left(cos\left(\frac{\theta}{2}\right), sin\left(\frac{\theta}{2}\right)\hat{\boldsymbol{n}}\right) \cdot (0, \vec{\boldsymbol{u}}) \cdot \left(cos\left(\frac{\theta}{2}\right), -sin\left(\frac{\theta}{2}\right)\hat{\boldsymbol{n}}\right) \tag{26}$$

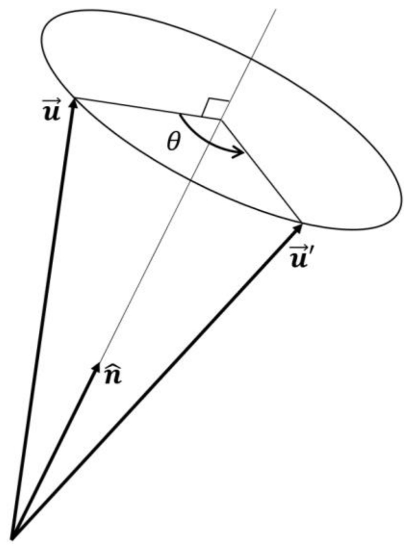

Figure 4-6. Rotation using quaternions

From (*26*), the rotation can be described as the product of the rotation matrix, $\boldsymbol{R} \in \mathbb{R}^{3\times3}$, and $\vec{\boldsymbol{u}}$.

$$\vec{\boldsymbol{u}}' = \boldsymbol{R} \cdot \vec{\boldsymbol{u}} = Rot\{\hat{\boldsymbol{n}}, \theta\} \cdot \vec{\boldsymbol{u}} \tag{27}$$

The conversion from quaternion to rotation matrix can be found in equation *(139)* in appendix A.1.

4.2.5. Gimbal lock

Gimbal lock is the loss of one rotational DoF due to the parallel configuration of two axes as a result of the rotations using Euler angles, as expressed in (*4*), (*5*), and (*6*) [12]. The gimbal lock happens when in a selected rotation representation, e.g., $Z - X' - Y''$, the second rotation angle equals 90 degrees, making the rotations about the first and third axes collinear. As the rotations are applied in a fixed order, the subsequent rotations act on the axis where preceding rotations have been already applied [13]. An expression for the gimbal lock can be seen using the equation (*7*). The $Z - X' - Y''$ representation creates a gimbal lock by replacing the values for the second rotation, α, by 90°:

$$ {}^{A}\boldsymbol{R}_{B} = \begin{bmatrix} cos\gamma \cdot cos\beta - sin\gamma \cdot sin\beta & 0 & cos\gamma \cdot sin\beta + sin\gamma \cdot cos\beta \\ sin\gamma \cdot cos\beta + cos\gamma \cdot sin\beta & 0 & sin\gamma \cdot sin\beta - cos\gamma \cdot cos\beta \\ 0 & 1 & 0 \end{bmatrix} \tag{28} $$

In this case, gimbal lock affects the Y''-axis rotation. It consists of some unreachable orientations around the Y''-axis that would be done around Z-axis instead.

The gimbal lock is an unavoidable problem while using Euler angles, but it can be faced with some known approaches. If the rotation around X'-axis is 90 degrees, but the rotation around other axes is not 90 degrees, the rotation order can be changed, such as the non-90-degrees rotation is placed in second position, using one of the possible combinations from the Tait-Bryan angles [14]. Another solution is to limit the second rotation to be 90 degrees. In this case, when it is detected that the value for the second rotation is in the range 90°±0.1°, it is set to 89.9°. This approximation avoids the expression (*28*), but it does not bring an accurate result. A complete solution to the gimbal lock problem is given by the use of quaternions, see section 4.2.4.

The results of this work are not flawed by the gimbal lock based on the following considerations. Equation (*87*) is the registration equation. Each transformation matrix in (*87*) does not include singularities. Some of these transformations come from the navigation system. The used surgical navigation system explained in section 4.3 gives the rotational part of the reported poses in quaternions, which are transformed to rotation matrices using equation *(139)* in appendix A.1. Another transformation comes from the digitally reconstructed radiograph (DRR) module. The DRR is implemented in section 5.6 and carries out its rotations using quaternions. The $Z - X' - Y''$ representation defines the DRR rotation. The standard patient reference frame is depicted in Figure 4-20. It is physically not possible to have 90° in the X'-axis as the C-arm will crash with the patient long before it reaches 90°; therefore, no gimbal lock occurs. Finally, the transformation matrix obtained as the registration result is not converted back to Euler angles, meaning the result is not affected by the gimbal lock.

4.3. Surgical Navigation Systems and Computer-Assisted Navigated Robots

One modern tool used inside operating theaters is the surgical navigation system. This device returns the position and orientation (pose) of a particular tool with respect to a static reference frame. Different types of technologies are used to build a navigation system. There are mechanical, electromagnetic and, optical tracking systems. In this work, optical tracking systems are used. In essence, they are high-precision stereo cameras. Figure 4-7 shows the used navigation system, the Polaris Vega manufactured by NDI.

Figure 4-7. Surgical navigation system NDI Vega

The static reference frame used as the coordinate system for the feedback poses is the camera reference frame. The objects of interest are points in the space that differ depending on the camera technology. Some manufacturers define a point as the intersection of a black and white checkered pattern. See Figure 4-8 left side. In the NDI navigation system, a point of interest is the center of a sphere coated with retroflected paint. This paint has a high reflection coefficient for infrared (IR) light. The system has an array of IR LEDs. The system cameras are sensitive to the IR spectrum and are installed behind an IR filter to reduce disturbance created by other light frequencies. When there is a sphere in the scene, it can be spotted as a bright circle on the image. To calculate not only position but also orientation, a set of points is required. The camera can track particular tools with unique geometric arrangements of three or more points of interest called rigid bodies (RB). In an RB, the relative position among points remains constant. An example of an RB made of retroflected spheres can be seen on the right side of Figure 4-8.

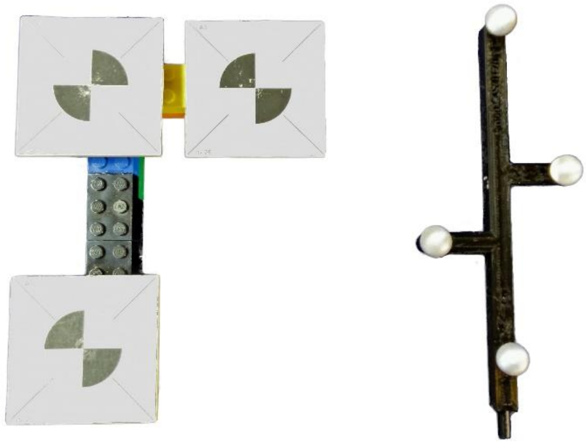

Figure 4-8. Example of rigid bodies (RB). On the left side, checkered-pattern RB. On the right, retroflected-sphere RB

The camera can compute the pose of the RBs on the scene using a definition file. This definition contains the layout of the points and a coordinate system used to describe each point. Making use of this definition file, the tracking system can compute the transformation between its reference frame and the RBs. The surgical navigation system reports the transformation of every loaded tool. A detailed explanation is given in section 5.1.1. A surgical navigation system can be used as a guide for the surgeon when using manual tools such as drill machines and reamers [15]. When the surgical navigation system is used not to guide a manual tool but a robot, the entire setup is classified as a computer-assisted navigated surgical robot system [16]. An RB is installed in one segment of the robot, commonly the last segment. The RB transformation with respect to the coordinate system of the robot tool is known. Another RB is attached to the patient's anatomy, close to where the surgical procedure will be carried out. This array of RBs allows the surgeon to use a planning-software to set an aiming pose for the robot tool, and the computer-assisted navigated robot is able to execute the procedure semi-automatically. Most modern systems include an additional linear guide rail parallel to the longest axis of the tool. The surgeon manipulates the movement on the linear guide rail. In other words, the robot holds the orientation and position of the tool, and the surgeon is in charge of inserting the tool into the patient. In this work, a six DoF robot from Adept Technology Inc is used. This robot has been adapted for spine surgery. The robot has a fine-tuning closed-motion control loop using the surgical navigation system. The robot aiming pose is given with respect to the navigation reference frame. Based on previous research, the robot guarantees repeatability similar to the navigation system precision, below 0.1mm [8]. See Figure 4-9.

Figure 4-9. Adept robotic surgical arms used in this work

It is worth clarifying that robot motion and control are beyond the scope of this work. The connection with the robot is limited to sending the aiming pose of the robot tool with respect to the patient's RB. On the other hand, finding the transformation of the patient's RB with respect to the pre-operative data is achieved through 2D to 3D registration, where all the efforts of this work are focused.

The following section explains the quality of screw insertion in the medical field based on the Gertzbein-Robbins scale.

4.4. Pedicle Screw Acceptance and Accuracy Assessment

The Gertzbein-Robbins scale evaluates the implant quality based on the screw lateral deviation compared with its pre-planned trajectory [9]. The pedicle screw deviation is known as breach, but it is also common to find the deviation under other names such as pedicle breach, screw breach, and cortical breach [17]. Since the implants are inserted through the pedicle until the vertebral body, canal encroachment must be avoided. See Figure 4-10.

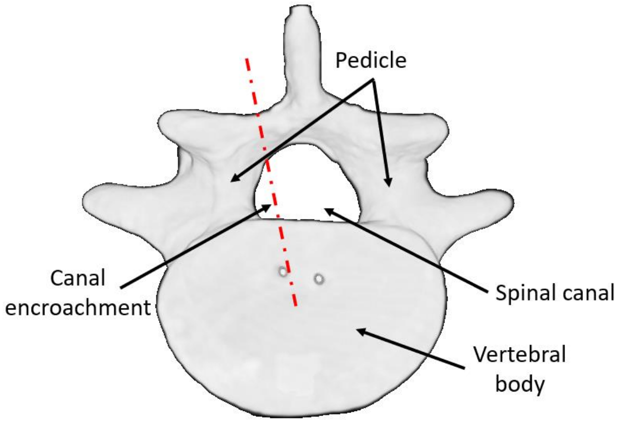

Figure 4-10. Lumbar vertebra components and canal encroachment clarification

Pedicle breach is calculated on an X-ray image after the implant insertion. It uses the mean deviation of the implant at both entry and exit position. The inserted implant deviation is defined as the magnitude of the perpendicular segment to the inserted implant trajectory that intersects the planned implant trajectory. The error calculation can be seen in Figure 4-11 and mathematically formulated in (*29*). The longitudinal error is ignored in the Gertzbein-Robbins evaluation since the screw length error is not a predictor of canal encroachment [18].

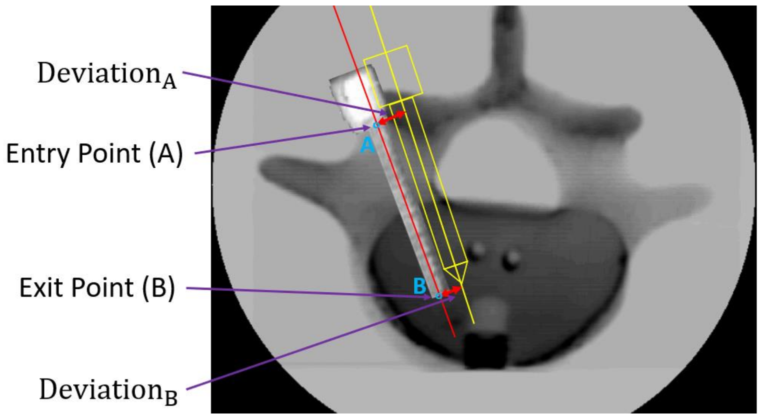

Figure 4-11. Points to calculate the implant breach, and deviation definition using the planned position and the inserted screw

$$Breach = \frac{Deviation_A + Deviation_B}{2} \tag{29}$$

where A is the entry point and B is the exit point [19].

An implant Grade A in the Gertzbein-Robbins scale indicates that the screw is inserted as planned, 0mm implant breach. Grade B indicates that the implant breach is less than 2mm. Grade C represents an implant breach <4mm; Grade D indicates a screw breach <6mm, and grade E in the Gertzbein-Robbins scale stands for a breach more than or equal to 6mm. Implants classified as Grade A and B are considered clinically acceptable while implants Grade C, D and E are subject to a risk of screw-related neurological complications [20]. One of the objectives of this work is to create a 2D/3D registration that leads to implant Grade A+B. It is shown in section 5.2.7 that the 2D to 3D registration cannot be evaluated directly. For this reason, using an implant as a measurement indirectly indicates the quality of the 2D to 3D registration.

In the following section, the mobile X-ray unit is discussed. This device plays a vital role in the registration process as it gives visual feedback on the position of the patient within the operating theater.

4.5. Mobile X-Ray Unit (C-Arm)

C-arm devices are the most common imaging-modalities inside the operating theater, used as guiding tools for controlling and monitoring surgical procedures [21]. C-arms allow the physician to obtain X-ray images of the patient along different projection directions. To this end, the device structure can be rotated and translated [22]. Although there are some C-arms with the capability of generating 3D volumes, they are mainly used to produce 2D X-ray images. In short terms, a C-arm consists of a primary structure in “C” shape, an image intensifier, and an X-ray tube (X-ray source). These last two components are installed at opposite ends of the main C-structure.

The image detector is the element that transduces the X-ray radiation into digital information. During the last decades, detectors have been improved to use less radiation, obtain better contrast, and offer higher image resolution. It is possible to discern between two technologies used in the detectors: X-ray image intensifiers and flat-panel detectors. Although both technologies create X-ray images, the results vary significantly between them. In Figure 4-12, it can be seen the change in the upper part of the C-arm depending on the technology. The left side of Figure 4-12 shows a C-arm with an image intensifier detector, and a C-arm with a flat-panel detector on the right side. From now on, X-ray image intensifiers will be referred as *image intensifiers*.

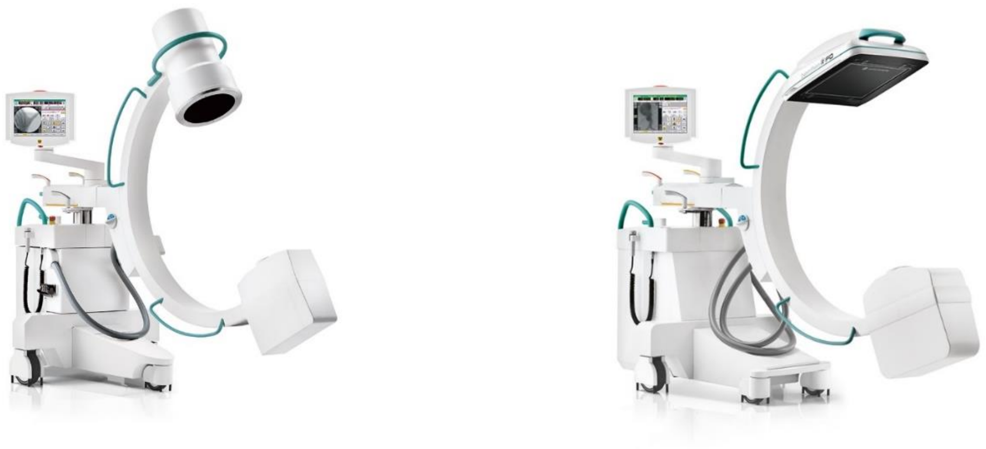

Figure 4-12. Mobile X-ray unit (C-Arm). Image intensifier on the left, flat-panel detector on the right
Courtesy of Ziehm Imaging GmbH [23]

4.5.1. Image Intensifier C-Arm

With the first commercial X-ray devices, a fluoroscopist was in the middle of the X-ray path to see the projection on a calcium-tungstate screen [24]. In the middle of the 20th century, the X-ray image intensifier was developed to create brighter X-ray images, which could also be recorded for further analysis. An image intensifier can be seen as a vacuum tower of four essential elements: input layer, focusing lenses, accelerating anode, and output layer. In the input layer, the X-rays are absorbed by a curved phosphor layer that converts the rays into light. Then a photocathode absorbs this light and turns it into electrons. The electrons are accelerated throughout the vacuum and focused on the output layer. The former operations are done by the accelerating anode and the focusing lenses, which are a set of negatively charged plates that repel the electrons. On the output layer, another phosphor layer, with different compounds as the input layer, transforms the electrons into visible light, which is absorbed by a camera (image detector) that generates the image [25]. The structure of the image intensifier can be seen in Figure 4-13.

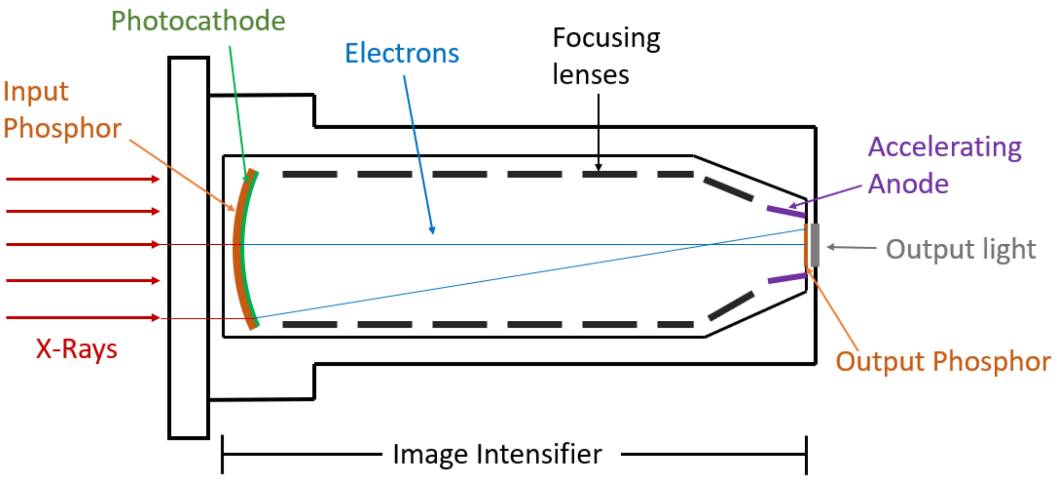

Figure 4-13. Structure of an image intensifier

It is worth noting that the input layer is curved to guarantee that the electrons travel the same distance from the input layer to the output layer. However, this projection from electrons going out from a curved surface to a flat output layer creates a pincushion distortion [26]. Other inherent problems associated with electro-optics produce S-shaped and spiral distortions [27]. One of the most significant contributors to these distortions is the earth's magnetic field, which is variant and complex to predict [28]. Thus, it is hardly possible to find a map that generalizes the distortion of an image intensifier C-arm. Instead, a particular distortion calibration must be achieved for each X-ray [29]. After the invention of the Flat-panel detector C-arm, which is explained in the following section, the image intensifier C-arm became known as conventional C-arm.

4.5.2. Flat-Panel Detector C-Arm

The image-creation process in the image intensifier C-arm is divided into four elements, while in the flat panel detector can be seen as two main components. In the first step, the X-ray radiation enters through a scintillation layer, transforming the radiation into direct light towards an array of detector elements. Each of these detector elements can compute the X-ray radiation on its surface, and then transduce this value into a grayscale level. In other words, each detector element gives the value of a pixel of the final image. The number of detector elements on the array determines the size of the X-ray image [30]. The working principle of the flat-panel detector brings several advantages compared with the image intensifier C-arm. The X-ray images are intrinsically free of distortion, it requires lower ionic radiation to create an image, and the image-forming process requires less energy than an image intensifier.

During the execution of this work, testing procedures and workflows were executed using an image intensifier C-arm. In some seldom trials, it was possible to access a flat-panel detector C-arm. Thus, this work considers the possibility of using an image for a C-arm with any technology. The procedures for C-arm parametrization (see section 5.4) and 2D to 3D registration (see section 5.1 and 5.2) expect X-ray images without distortion. When using images from a conventional C-arm, the X-ray images are processed by an undistortion method explained in chapter 5.3.

4.5.3. Mathematical Model of the C-Arm

As explained before, a C-arm is a mobile X-ray device. It can be seen as a camera with an image sensor sensitive to X-ray radiation. The C-arm requires an X-ray source, which is usually referred to as an X-ray tube.

A *camera* can be defined as a mapping process between the 3D world and a 2D image. The most basic mathematical model of the camera is the *pinhole camera model*, mainly applied to charge-coupled device

(CCD) cameras [31]. However, some studies have found that X-ray and C-arm devices also satisfy the pinhole camera model [32] [33].

The pinhole camera model is based on the camera obscura device that was first used to take pictures [34]. It consists of two parallel screens, as can be seen in Figure 4-14. In the first screen, there is a pinhole, and through it, some rays pass and form an image on the second screen, which will be called the *retinal plane*. All the rays that form the image are coming from the pinhole. Therefore, this point is called the *optical center* ($\boldsymbol{c}$). The screen, where the optical center is located, is called the *focal plane*, and the distance f from the retinal plate to the optical center, is called the *focal length* of the optical system.

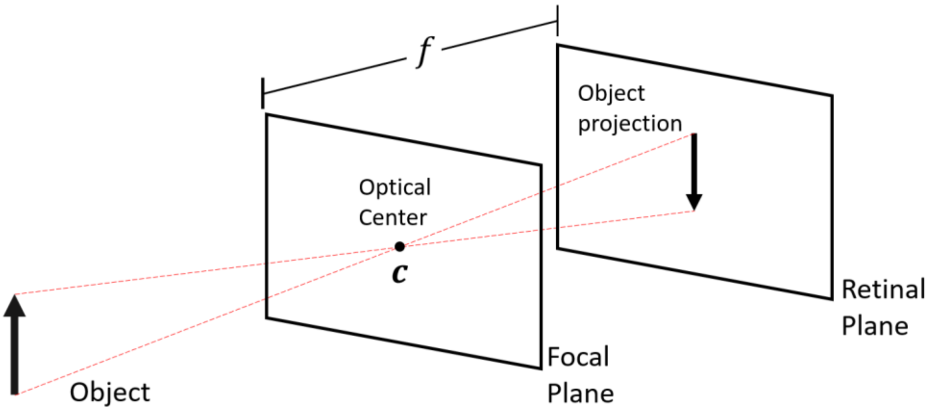

Figure 4-14. Pinhole camera model

The projection from a perpendicular ray from $\boldsymbol{c}$ to the retinal plane can be seen in Figure 4-15 as the point $\boldsymbol{c}_r$. In an ideal case, the point $\boldsymbol{c}_r$ and the image origin are aligned. In the general case, there is a displacement, and the point $\boldsymbol{c}_r$ has a coordinate (u_0, v_0) with respect to the image origin.

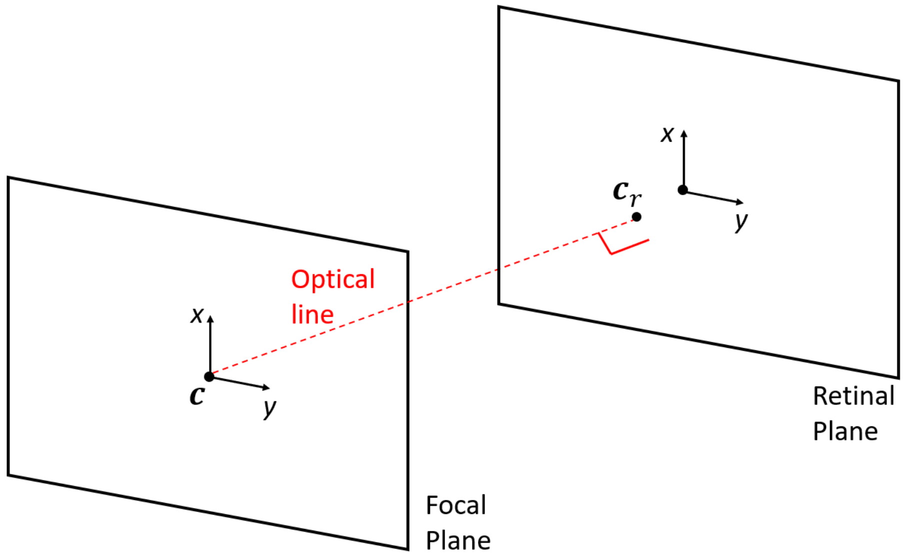

Figure 4-15. Optical Line, and image origin of the pinhole camera model

Let the origin of a Euclidian coordinate system be centered on the optical center (point $\boldsymbol{c}$), and the equation $z = f$ defines the retinal plane. A point $\boldsymbol{x}_0 = (x, y, z)^T$ can be projected into the retinal plane by a ray going through the optical center, as depicted in Figure 4-16.

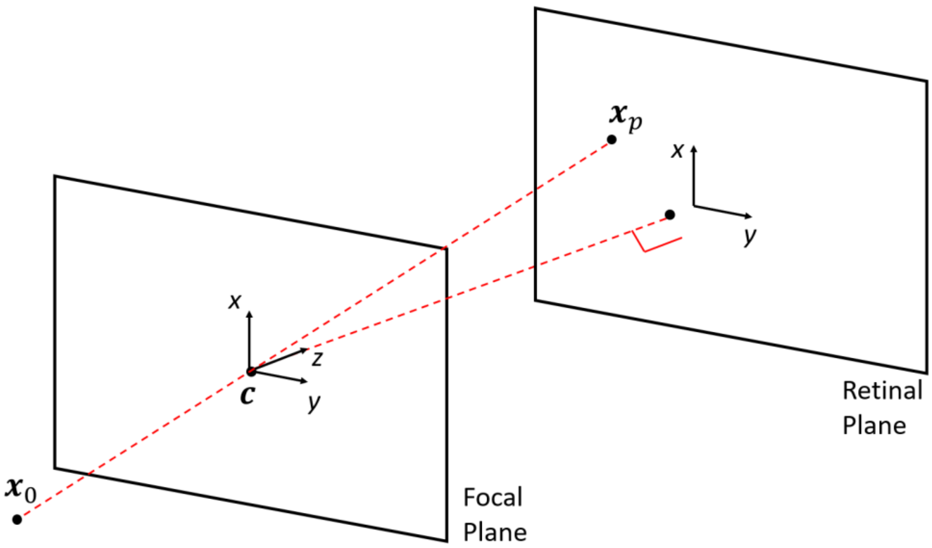

Figure 4-16. Projection of Point $\boldsymbol{x}_0 \in \mathbb{R}^3 \longrightarrow \boldsymbol{x}_p \in \mathbb{R}^2$

Using similar triangles, the mapping from $\boldsymbol{x}_0 \in \mathbb{R}^3 \longrightarrow \boldsymbol{x}_p \in \mathbb{R}^2$ is computed as $(x, y, z)^T \longrightarrow \left(f \cdot \frac{x}{z}, f \cdot \frac{y}{z}\right)^T$.

This projection can also be described using real projective coordinates, with the center of the reference frame in the optical center (point $\boldsymbol{c}$), where a point in $\wp^4$ is projected to $\wp^3$. This homogenous reference frame is called the camera *coordinate system* {CCS}, and the optical center is called the *camera coordinate origin*. The Euclidian point $\boldsymbol{x}_0$ is represented in $\wp^4$ as $\widetilde{\boldsymbol{x}}_0 = (x, y, z, 1)^T$, and the projected $\mathbb{R}^2$ point $\boldsymbol{x}_p$ is represented in $\wp^3$ as $\widetilde{\boldsymbol{x}}_p = (f \cdot x, f \cdot y, z)^T$.

The following transformation defines the projection $\wp^4 \longrightarrow \wp^3$:

$$\begin{bmatrix} f \cdot x \\ f \cdot y \\ z \end{bmatrix} = \begin{bmatrix} f & 0 & 0 & 0 \\ 0 & f & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix} \tag{30}$$

As stated before, often there is a displacement of the optical center with respect to the image center. In such case equation *(30)* can be expressed as follows:

$$\begin{bmatrix} f \cdot x + z \cdot u_0 \\ f \cdot y + z \cdot v_0 \\ z \end{bmatrix} = \begin{bmatrix} f & 0 & u_0 & 0 \\ 0 & f & v_0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix} \tag{31}$$

The matrix in *(31)* is known as the *camera projection matrix* $\boldsymbol{P}$, and, thus, *(31)* can be rewritten as:

$$\widetilde{\boldsymbol{x}}_p = \boldsymbol{P} \cdot \widetilde{\boldsymbol{x}}_0 \tag{32}$$

Now, it is useful to include a transformation from the camera coordinate frame to a *world coordinate system* {WCS}. The camera coordinate frame will be then redefined, with respect to the new world coordinate frame. To do so, let define the rotation matrix $\boldsymbol{R}$ and the translation vector $\vec{\boldsymbol{c}}$, which describe the Euclidean transformation from {WCS} to {CCS}. A point $\tilde{\boldsymbol{x}}_{wcs} \in \wp^4$ in the {WCS} can be projected to $\tilde{\boldsymbol{x}}_{p_{wcs}} \in \wp^3$ expanding the equation (*32*) as follows:

$$\tilde{\boldsymbol{x}}_{p_{wcs}} = \begin{bmatrix} f & 0 & u_0 & 0 \\ 0 & f & v_0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \cdot \begin{bmatrix} \boldsymbol{R} & -\boldsymbol{R} \cdot \vec{\boldsymbol{c}} \\ \vec{\boldsymbol{0}}^T & 1 \end{bmatrix} \tilde{\boldsymbol{x}}_{wcs} \tag{33}$$

From equation (*33*), it can be seen that the matrix $\boldsymbol{P}$ can be updated to:

$$\boldsymbol{P} = \begin{bmatrix} f & 0 & u_0 & 0 \\ 0 & f & v_0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} \boldsymbol{R} & \vec{\boldsymbol{0}} \\ \vec{\boldsymbol{0}}^T & 1 \end{bmatrix} \begin{bmatrix} \boldsymbol{I} & -\vec{\boldsymbol{c}} \\ \vec{\boldsymbol{0}}^T & 1 \end{bmatrix} \tag{34}$$

The matrix $\boldsymbol{P}$ is then a composition of an upper-triangular matrix and a transformation matrix. The coefficients of the upper-triangular matrix depend only on the building properties of the camera, so it is called the *intrinsic parameters matrix* ($\boldsymbol{I_p}$). The second part depends on an external world coordinate, which can be changed. The second part is then called the *extrinsic parameter matrix* ($\boldsymbol{E_p}$).

Extrinsic and intrinsic parameters

When talking about cameras, the $\mathbb{R}^3 \longrightarrow \mathbb{R}^2$ mapping decreases the space dimension and converts length units into pixel units. The unit conversion is done using factors. Let these factors be m_x for the x-direction and m_y for the y-direction. The focal length will be represented in pixel dimensions by α_x and α_y, and the displacement of the image origin in pixel units is represented by (u_x, v_y). Making use of these factors the intrinsic matrix can be converted from length domain to pixel domain, obtaining:

$$\boldsymbol{I_p} = \begin{bmatrix} fm_x & 0 & u_0 m_x & 0 \\ 0 & fm_y & v_0 m_y & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} = \begin{bmatrix} \alpha_x & 0 & u_x & 0 \\ 0 & \alpha_y & v_y & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \tag{35}$$

There is still one parameter missing to create the general case of the intrinsic matrix, the *skew parameter* s. The main axes of the retinal coordinate system may be non-orthogonal, so the skew parameter (ideally zero) gives the cosine of the angle between the retinal axes. All in all, the intrinsic parameters would be written as:

$$\boldsymbol{I_p} = \begin{bmatrix} \alpha_x & s & u_x & 0 \\ 0 & \alpha_y & v_y & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \tag{36}$$

On the other hand, the extrinsic parameters can be defined as a transformation matrix defining the new reference of the camera system:

$$\boldsymbol{E_p} = \begin{bmatrix} \boldsymbol{R} & -\boldsymbol{R} \cdot \vec{\boldsymbol{c}} \\ \vec{\boldsymbol{0}}^T & 1 \end{bmatrix} = \begin{bmatrix} \boldsymbol{R} & \vec{\boldsymbol{0}} \\ \vec{\boldsymbol{0}}^T & 1 \end{bmatrix} \cdot \begin{bmatrix} \boldsymbol{I} & -\vec{\boldsymbol{c}} \\ \vec{\boldsymbol{0}}^T & 1 \end{bmatrix} = \boldsymbol{R}[\boldsymbol{I} \quad | \quad -\vec{\boldsymbol{c}}] \tag{37}$$

Finally, the general form of the camera projection matrix is given in the following equation:

$$\boldsymbol{P} = \boldsymbol{I_p} \cdot \boldsymbol{E_p} = \begin{bmatrix} p_{11} & p_{12} & p_{13} & p_{14} \\ p_{21} & p_{22} & p_{23} & p_{24} \\ p_{31} & p_{32} & p_{33} & p_{34} \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{38}$$

Estimating the camera projection is carried out by a camera calibration procedure, which is the topic of section 4.5.4.

Units of the intrinsic and extrinsic parameters

Before going ahead with the C-arm calibration, the units of the parameters in the matrices would be clarified. Starting with the extrinsic matrix $\boldsymbol{E_p}$, the elements of $\boldsymbol{R}$ are dimensionless since it is a rotation matrix [35]. Each of the elements of the vector $\vec{\boldsymbol{c}}$ are in mm. To find the dimensions of $\boldsymbol{I_p}$, the following equality will be enforced $\boldsymbol{P} = \boldsymbol{I_p}$, which happens naturally when $\boldsymbol{E_p}$ is the identity matrix, i.e., no extrinsic rotation nor translation.

From equation (*32*) and (*36*), it is possible to find that:

$$\begin{bmatrix} x'_p \\ y'_p \\ z'_p \end{bmatrix} = \begin{bmatrix} \alpha_x & s & u_x & 0 \\ 0 & \alpha_y & v_y & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix} \tag{39}$$

Developing every component of the projection, it can be seen that:

$$\begin{aligned} x'_p &= \alpha_x \cdot x + s \cdot y + u_x \cdot z \\ y'_p &= \alpha_y \cdot y + v_y \cdot z \\ z'_p &= z \end{aligned} \tag{40}$$

With $x_p = \frac{x'_p}{z'_p}$ and $y_p = \frac{y'_p}{z'_p}$, the equation (*40*) reads

$$\begin{aligned} x_p &= \alpha_x \cdot \frac{x}{z} + s \cdot \frac{y}{z} + u_x \\ y_p &= \alpha_y \cdot \frac{y}{z} + v_y \end{aligned} \tag{41}$$

x_p and y_p are points on the X-ray image, so their dimension is in pixels. The coordinates $x, y,$ and z are in millimeters. From equation (*41*), it can be seen that α_x, α_y and s have pixel dimensions. From equation (*35*), it can be noticed that α_x, α_y, u_x and v_y contain a m_x and m_y component. As this process transforms from millimeters to pixels, m_x and m_y are seen with pixel/mm units and can be physically associated with the detector pixel size. The quantities f, u_0 and v_0 are in accordance with the pinhole camera model in millimeters.

4.5.4. C-Arm Calibration Using the Pinhole Camera Model

This procedure has been studied previously in the literature [36] [37] [38]. In this work, the C-arm is calibrated using the direct linear transformation (DLT) algorithm [31]. It requires at least six correspondences $\tilde{\boldsymbol{x}}_i \rightarrow \tilde{\boldsymbol{x}}_{pi}$,i.e., having the homogenous coordinates of at least six points in $\wp^4$ and their locations on the projected $\wp^3$space.

From equation (*32*) using the general form of $\boldsymbol{P}$ in equation (*38*), it can be found that the point $\tilde{\boldsymbol{x}}_i = (x_i, y_i, z_i, 1)^T$ has a projection $\tilde{\boldsymbol{x}}_{pi} = \left(x_{pi}, y_{pi}\right)^T$ that follows the next relation:

$$
\begin{aligned}
x_{pi} &= \frac{p_{11} \cdot x_i + p_{12} \cdot y_i + p_{13} \cdot z_i + p_{14}}{p_{31} . x_i + p_{32} \cdot y_i + p_{33} . z_i + p_{34}} \\
y_{pi} &= \frac{p_{21} \cdot x_i + p_{22} \cdot y_i + p_{23} \cdot z_i + p_{24}}{p_{31} \cdot x_i + p_{32} \cdot y_i + p_{33} \cdot z_i + p_{34}}
\end{aligned}
\tag{42}
$$

Solving these relations for the $\boldsymbol{P}$ coefficients, it can be found that:

$$
\begin{aligned}
p_{11} \cdot x_i + p_{12} \cdot y_i + p_{13} \cdot z_i + p_{14} - p_{31} \cdot x_{pi} \cdot x_i - p_{32} \cdot x_{pi} \cdot y_i - p_{33} \cdot x_{pi} \cdot z_i - x_{pi} \cdot p_{34} = 0 \\
p_{21} \cdot x_i + p_{22} \cdot y_i + p_{23} \cdot z_i + p_{24} - p_{31} \cdot y_{pi} \cdot x_i - p_{32} \cdot y_{pi} \cdot y_i - p_{33} \cdot y_{pi} \cdot z_i - y_{pi} \cdot p_{34} = 0
\end{aligned}
\tag{43}
$$

So, for each pair of points, there are two equations with 12 unknowns; for this reason, it is required to have a minimum of six corresponding points. A homogenous system of equations of the form $\boldsymbol{A} \cdot \vec{\boldsymbol{q}}_{\boldsymbol{p}} = \vec{\boldsymbol{0}}$ can be obtained with six points, but the general case with *n* points is described here. Using a least square solution of homogenous system of linear equations, see Appendix A.6, $\vec{\boldsymbol{q}}_{\boldsymbol{p}}$ can be found as the unit singular vector of $\boldsymbol{A}$, which corresponds to the smallest singular value [39].

The system $\boldsymbol{A} \cdot \vec{\boldsymbol{q}}_{\boldsymbol{p}} = \vec{\boldsymbol{0}}$ has the form:

$$
\begin{bmatrix}
x_1 & y_1 & z_1 & 1 & 0 & 0 & 0 & 0 & x_{p1}X_1 & x_{p1}Y_1 & x_{p1}Z_1 & x_{p1} \\
0 & 0 & 0 & 0 & x_1 & y_1 & z_1 & 1 & y_{p1}X_1 & y_{p1}Y_1 & y_{p1}Z_1 & y_{p1} \\
\vdots & & \ddots & & & & & & & & & \vdots \\
\vdots & & & \ddots & & & & & & & & \vdots \\
\vdots & & & & \ddots & & & & & & & \vdots \\
\vdots & & & & & \ddots & & & & & & \vdots \\
\vdots & & & & & & \ddots & & & & & \vdots \\
\vdots & & & & & & & \ddots & & & & \vdots \\
\vdots & & & & & & & & \ddots & & & \vdots \\
\vdots & & & & & & & & & \ddots & & \vdots \\
x_n & y_n & z_n & 1 & 0 & 0 & 0 & 0 & x_{pn}x_n & x_{pn}y_n & x_{pn}z_n & x_{pn} \\
0 & 0 & 0 & 0 & x_n & y_n & z_n & 1 & y_{pn}x_n & y_{pn}y_n & y_{pn}z_n & y_{pn}
\end{bmatrix}
\begin{bmatrix}
p_{11} \\ p_{12} \\ p_{13} \\ p_{14} \\ p_{21} \\ p_{22} \\ p_{23} \\ p_{24} \\ p_{31} \\ p_{32} \\ p_{33} \\ p_{34}
\end{bmatrix}
=
\begin{bmatrix}
0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0
\end{bmatrix}
\tag{44}
$$

$\vec{\boldsymbol{q}}_{\boldsymbol{p}}$ contains the 12 coefficients of the $\boldsymbol{P}$ matrix. However, for this work, it is more useful to represent the decomposed matrix $\boldsymbol{P}$ in its intrinsic and extrinsic parameters, i.e., matrix $\boldsymbol{I}_{\boldsymbol{p}}$ and $\boldsymbol{E}_{\boldsymbol{p}}$. This decomposition is done through an RQ-decomposition, which is the product of right-triangular matrix ($\boldsymbol{I}_{\boldsymbol{p}}$) and an orthogonal matrix ($\boldsymbol{E}_{\boldsymbol{p}}$). It is common to find in literature a QR-factorization method like Gram-Schmidt or Householder, but it does not apply for this case as the required factorization is the RQ. The process to calculate the RQ-decomposition is explained in Appendix A.4.

The 12 non-zero coefficients of the matrix $\boldsymbol{P}$ can be taken as a 3x4 matrix, that can be split in a 3x3 matrix called $\boldsymbol{P}_{\boldsymbol{R}}$ and a column vector $\vec{\boldsymbol{p}}_{\boldsymbol{C}}$, such as:

$$\boldsymbol{P} = [\boldsymbol{P_R} \quad \vec{\boldsymbol{p}}_{\boldsymbol{C}}] \tag{45}$$

From equation (*45*), the $\boldsymbol{P_R}$ matrix is factorized using RQ-decomposition. The result is the product $\boldsymbol{I_p} \cdot \boldsymbol{R}$. The translation of the extrinsic matrix, i.e., point $\boldsymbol{c} = (x_c, y_c, z_c)^T$ can be found solving the following equation:

$$\boldsymbol{c} = -\boldsymbol{P}_{\boldsymbol{R}}^{-1} \cdot \vec{\boldsymbol{p}}_{\boldsymbol{C}} \tag{46}$$

The proof of equation (*46*) is out of the scope of this work. However, it can be illustrated with a geometrical interpretation [40]. Three non-zero rows form the $\boldsymbol{P}$ matrix. The third row describes the focal plane, and the first and second rows describe two normal planes, whose intersect with the retinal plane forms its X- and Y- axes. The intersection of those two planes is the line, which goes from the optical center (point $\boldsymbol{c}$) to the origin of the image in the retinal plane (point $\boldsymbol{c}_0$). The intersection of these three planes defines the optical center of the camera, which is on the focal plane, as Figure 4-17 shows. Equation (*46*) calculates the intersection of the three planes that compose the matrix $\boldsymbol{P}$ and then finds the optical center.

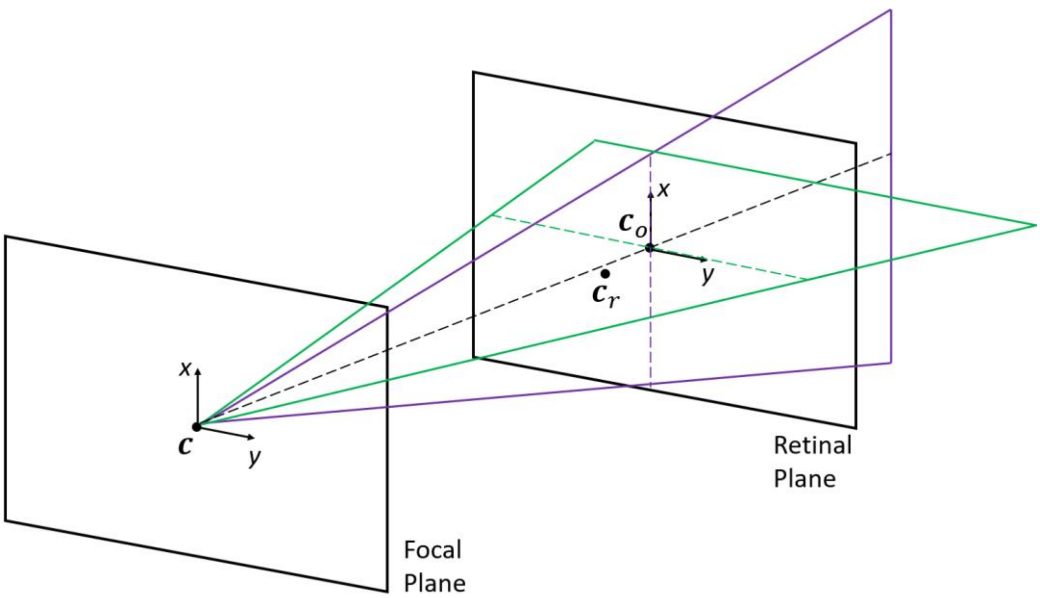

Figure 4-17. Geometrical Interpretation of the row vectors in matrix $\boldsymbol{P}$

4.6. Registration Procedures for Images from Different Modalities

Imaging registration is a standard procedure not only in medical applications but also in robotics and computer vision. It consists of finding the transformation between the reference frames of two images regardless of their dimensionality, i.e., 2D or 3D [41]. The classification of the registration is based on nine criteria, and they are dimensionality, nature of registration basis, elasticity of the transformations, domain of the transformations, interaction, optimization procedure, modalities involved, subject, and object [42]. These criteria work as a guideline for implementing the most suitable registration procedure in a specific application. The primary goal of the imaging registration is finding the best alignment of one image against the other. When using an iterative approached, the goal is achieved by comparing the improvement reached in the current stage with respect to a previous state. This process consists of three components: problem statement, registration model, and optimizer. The three registration components implicitly describe the former nine criteria [43]. The following classification criteria are elaborated in the next sections: image modalities, image dimensionality, feature basis, and processing level. They are worthy of elaborating as they are the basis for finding a suitable implementation for this work.

4.6.1. Imaging Modalities for Generation of 2D and 3D Image Data

In this section, an introduction to 2D and 3D modalities is presented. They are sorted by order of appearance in a surgical workflow, i.e., pre-operative modalities: image modalities taken before the surgery; and intra-operative modalities: image modalities taken during the surgical procedure.

Pre-Operative Modalities

Pre-operative modalities are medical imaging used for the diagnosis of patients and planning surgical interventions. Such modalities have a top-notch resolution and image quality and are commonly found outside of the operating room. Pre-operative modalities can also be defined as imaging modalities taken before the medical procedure but inside the operating room. For this work, the elapsed time and location of the modality acquisition are not included in the definition. Only imaging quality for the correct diagnosis and planning, and acquisition before the medical procedure are considered for the classification. Depending on the nature of the intervention, pre-operative modalities can be 2- or 3-dimensional (2D or 3D) [44]. In spine surgery, 3D modalities are preferred. A 3D imaging modality of the spine structure gives a better overview of the complex vertebra anatomy, which offers better detail for diagnosis purposes. For planning, deviations on planned pedicle screws can compromise screw fixation, spinal nerves, and visceral organs during executing a non-invasive procedure due to lack of spatial information [45] [46]. For the reasons mentioned above, this work is concentrated only on 3D pre-operative modalities.

Magnetic Resonance Imaging (MRI)

Magnetic resonance imaging devices create a 3D image of a biological specimen. MRI devices rely on the principle of nuclear magnetic resonance, which establishes a relation of energy absorption of a specific nucleus within an electromagnetic field. Human beings are made up to 60% of water. MRI devices are calibrated with the hydrogen nuclei energy absorption taking advantage of the large amount of H_2O molecules in the human body [47]. Using strong magnetic fields, up to 8 teslas, MRI scanners can create 3D images with high resolution and high contrast comprising different types of soft tissues [48]. As its working principle is based on electromagnetism, the subject is not exposed to ionizing radiation during data acquisition [49]. The ability to generate high-quality images without using ionizing radiation is the most significant advantage of the MRI devices. An MRI device is depicted in Figure 4-18.

Figure 4-18. Magnetic Resonance Imaging Device
By KasugaHuang, 2006, https://commons.wikimedia.org/wiki/File:Modern_3T_MRI.JPG. Used under Creative Commons Attribution 2.5 Generic License: https://creativecommons.org/licenses/by-sa/2.5/deed.en

CT- Scan (Computed Tomography Scan)

CT-scans devices are also able to create 3D volumes from an object. The difference with respect to an MRI is the working principle. CT-scans use ionizing radiation, like an X-ray, which gives better contrast for bone structure. Building a 3D model from a set of 2D images was first demonstrated in 1917 by J. Radon, who proved that a 3D object can be reconstructed by an infinite number of its projections [50] It is out of the scope of this work to clarify the CT-scan image creation process, but it is found in the literature that the number of images required to create a 3D volume in a conventional CT is about 180 projections [51] [52]. Modern CT-scans work based on the same fundamental principle, but every year the technology improves, enabling results with better contrast and spatial resolution, and reducing the radiation dose [53]. A CT-scanner is depicted in Figure 4-19.

Figure 4-19. Computed Tomography Scanner
By daveynin, 2012, https://commons.wikimedia.org/wiki/File:UPMCEast_CTscan.jpg. Used under Creative Commons Attribution 2.0 Generic License: https://creativecommons.org/licenses/by/2.0/deed.en

CT-scans demonstrate higher sensitivity to bone structure than MRIs, which helps to discriminate better bone from soft tissue and, consequently, achieve a better registration [54] [55]. In medical guidelines for trauma in the spine, CT-scans are suggested as primary screening for injured patients [56]. Additionally, registering X-ray imaging with MRI has reported inconvenient for finding similarities due to differences in tissue contrast [57] [58]. There are some studies reporting ways to overcome the lack of similarities in the image information, but the reported registration errors are higher than the expected to achieve in this work [59]. In other cases, there has been attempts to create a pseudo-CT from the MRI, but it leads to a higher error when compared to a registration with a CT [60]. For this reason, CT-scan imaging is used as the 3D pre-operative modality.

Intra-Operative Modalities

Those are modalities used at the moment of the medical intervention. Therefore, they are found inside the operating room. Although it is also possible to use CT-scans as intra-operative modalities, it is rare to find a CT-scanner inside an operating theater. It is also possible to find ultrasound imaging as a 3D intra-operative modality. However, it is uncommon to use this modality in spine surgery for pedicle screw fixation because it creates imaging artifacts, and bone boundaries appear some millimeter thicker [61]. Therefore, 2D images as the intra-operative modality are expected during the development of this work, specifically X-ray images coming from a C-arm. As explained in chapter 4.5, C-arm devices are ubiquitously found inside operating theaters. Consequently, using images from C-arms as the default intra-operative modality makes it the most logical and practical solution for a real-life application.

Digital Imaging and Communications in Medicine (DICOM)

DICOM is the standard for managing medical imaging data. It is not just an image or file format. It is an all-encompassing data transfer, storage, and display protocol built and designed to cover all functional aspects of digital medical imaging [62]. DICOM files store both data and metadata of the images. The image itself is considered the data and is the core of the information, but the metadata, e.g., width, height, bits per pixel, is also part of the DICOM files.

The *DICOM Data Dictionary* is the registry of all standard data items (attributes) used in digital medicine. The list contains more than 2,000 items, which are organized in groups. The group number and the element number of each item form its tag. CT images are organized in the DICOM standard as a set of slices. A slice is a 2D image composed of rows and columns with a certain thickness. Some necessary information about the pixels and slices of a CT file is in Table 4-1, reproduced from the original “Table C.7-10. Image Plane Module Attributes” of DICOM object definition [63].

Table 4-1. Relevant DICOM Tags for CT-scans

Attribute Name	Tag	Attribute Description
Pixel Spacing	(0028,0030)	Physical distance in the patient between the center of each pixel, specified by a numeric pair - adjacent row spacing (delimiter) adjacent column spacing in mm.
Image Orientation (Patient)	(0020,0037)	The direction cosines of the first row and the first column with respect to the patient.
Image Position (Patient)	(0020,0032)	The x, y, and z coordinates of the upper left-hand corner (center of the first voxel transmitted) of the image, in mm.
Slice Thickness	(0018,0050)	Nominal slice thickness, in mm.
Slice Location	(0020,1041)	Relative position of the image plane expressed in mm

The standard definition of the anatomical orientation frame in a DICOM image is given in the section "C.7.6.2.1.1 Image Position and Image Orientation" of the DICOM object definition [64]. It states that the X-axis increases towards the left-hand side of the patient. The Y-axis increases to the patient's posterior side, and the Z-axis increases toward the patients' head. The previous definition can be seen in Figure 4-20.

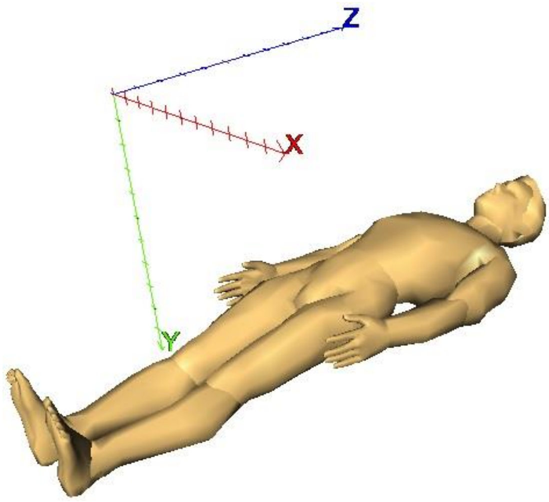

Figure 4-20. Anatomical orientation of the DICOM reference frame

4.6.2. Registration Based on Image Dimensionality

3D to 3D Registration

The 3D to 3D registration, also called 3D/3D registration, consists of finding the transformation in $\mathbb{R}^3$ between reference frames of one 3D dataset to another 3D dataset. The 3D/3D registration is classified between two different approaches, rigid and non-rigid. In rigid registration, it is assumed that there is no deformation between models, so six DoF are enough to represent the registration. In the end, the result is a rigid transformation. On the other hand, non-rigid transformations require more parameters to characterize the deformation between 3D sets. This non-rigid behavior can be described by applying sheering, projective, and deformable transformations [65]. The result of a non-rigid registration is an affine transformation, which includes the sheering and projective effects, and a non-linear transformation matrix for the deformation.

The selection of the 3D/3D registration approach is mostly based on the body segment being 3D/3D registered. For instance, in the case of soft tissues, non-rigid registration is suitable for changes caused by patient's movements or manipulation during the procedure [66]. Registrations in orthopedic applications allow using rigid transformations. There is no significant deformation in bone structure between pre- and intra-operative modalities, so a rigid approach accurately describes the transformation between volumes.

2D to 2D Registration

For 2D to 2D (2D/2D) registration, it is expected to find a transformation matrix in $\mathbb{R}^2$, which defines the transformation from the reference frames of a 2D image to another. As in the 3D/3D registration, it has rigid and non-rigid variants. However, the reduction of dimensionality gives to this registration a complexity of an order of magnitude less than the 3D case [43]. 2D/2D registration is useful for registering different modalities or following the evolution of treatment [67].

2D to 3D Registration

The result of a 2D to 3D registration, also called 2D/3D registration or 3D/2D registration, is a transformation matrix in $\mathbb{R}^3$. This matrix gives the spatial alignment of the 3D data to the projective plane of the 2D data. This procedure can be executed in two different ways. One approach requires a set of 2D images, which are used to build a 3D volume. Then a 3D/3D registration is executed using the pre-operative 3D volume and the built 3D volume. The obtained result gives the pose of pre-operative data with respect to the created volume. This approach has the drawback that creating a good quality volume requires a large set of images, e.g., between 50 to 100 [68]. The need for many X-ray acquisitions shifts this approach away from the main objective of this work, which is using as low ionizing radiation as possible.

A second approach consists of creating projections from different poses of the 3D volume on a 2D plane. These projected images are compared with the original X-rays. The idea with this approach is to find the 3D volume pose that makes the projected and original 2D images look the most alike [58]. This approach requires as few as one X-ray, but it works optimally with two [69]. Some studies suggest that increasing the number of X-rays by more than two does not improve the quality of the registration as long as the two used X-rays are orthogonal [70]. Based on the amount of radiation required, the second approach is considered the most suitable for this work.

4.6.3. Feature Basis for Registration

2D to 3D registration relies on finding similarities between unique features between the 2D and the 3D modalities. Another methodologic division consists of determining the type of used features. It is possible to use artificial elements attached to the body segment to find the registration. This approach is classified as extrinsic registration. One more way involves finding a relation between modalities using the natural elements of the human body. When the registration relays only on patient features, the registration is called intrinsic registration.

Extrinsic Methods

This method relies on artificial objects that can be seen by the 3D and 2D imaging systems, e.g., intra-muscular beads, fiducials on the skin, or a rigid reference attached to bone structure [71]. Since these artificial objects can be observed in both modalities and are distinguishable from human elements, this process can be automated without significant difficulties [72]. When rigid frames are used for extrinsic registration, the registration is so accurate that it was considered for a long time, the gold standard [73] [74] [75]. Intra-muscular beads and skin fiducials are less traumatic for the patient, but the accuracy is lower than with using rigid frames [76]. Extrinsic methods have shown a high accuracy under low computational power [77]. The biggest drawback of these methods is that they rely on external elements attached to the patient since the pre-operative modality is taken [78]. As reducing the amount of additional external object is wanted for the development of this work, this method is discarded. Therefore, extrinsic methods are not going to be considered further on.

Intrinsic Methods

On the other hand, intrinsic methods rely only on the patient anatomical structures captured by the modalities. As a difference with the extrinsic methods, there is no predefined landmark to be used as the base point for the registration [79]. Possible landmarks are bone edges or other anatomical features, and then the registration is further classified as feature-based method.

When using X-ray images, the elements with better visualization are bone structures. Methods using the gradients of bone edges in 2D images, or gradients of bone surfaces in the 3D case as the base for registration are known as gradient-based methods. There is a third variant called intensity-based method, which relays only on the pixel and voxel intensities in the different modalities. The division of the intrinsic methods is further elaborated in this works as processing methods for registration.

4.6.4. Processing Methods for Registration

From now on, all approaches that can be taken for executing a registration using intrinsic methods will be referred to as processing methods. The following methods will be explained in detail below: feature-based, gradient-based, and intensity-based methods.

Feature-Based Methods

Feature-based methods aim at finding geometrical entities like isolated points, point sets, curves, contours, or surfaces. In general, 3D models are transformed into a silhouette, while 2D images are transformed into contours. Once the features are selected in both modalities, the registration runs by projecting the 3D modality features in different poses into a 2D plane. The registration is achieved when the minimum distance between the 2D image contour and the projected image contour is found [80]. In the case of bone structure, this approach is considered attractive because they have a well-defined contour [81]. This procedure works best when the modalities are previously segmented and later geometrical features extracted. However, segmenting 2D and 3D images is not a trivial process. Poor segmentations directly impact the result of the 2D/3D registration, so robust segmentation methods must be implemented [82]. There are some approaches to overcome this problem using manual segmentation, but they can be tiresome and prone to error depending on the user's skills [83]. An additional problem arises as unmatched features appear, i.e., features that appear only in one modality. These features are not used for the registration and do not have an effect on the registration error, but they must be handled carefully to avoid mismatching [84]. Some solutions to the problems of using feature-based methods are found in the literature [85] [86] [87] [88] [89]. Nevertheless, this work will focus on other methods that have shown better accuracy in spine applications [90] [91].

Gradient-Based Methods

Gradient-based methods work considering the gradients of surfaces and contours. From a geometrical point of view, it is known that both straight lines and planes can be characterized by their corresponding normal vectors. In the case of contours and surfaces, normal vectors change at every point of the geometry due to its non-zero curvature. Gradients are more intense at the borders of anatomical structures [92]. The gradient-based method creates a gradient image of the X-ray and a 3D gradient volume from the 3D modality. Later, the projection of the 3D gradient volume generates a 2D gradient image, which is compared with the gradient created out of the X-ray [93]. Projections of different directions are tried out and the registration is attained by finding the one that yields the best alignment between the projected image and the X-ray gradient image [94].

Intensity-Based Methods

These registration methods rely on the intensity in pixels of 2D images and voxels of 3D images [95]. Specifically, the voxels in the volume are projected into a 2D plane using a ray-casting approach called a *digitally reconstructed radiograph* (DRR). This projection depends on the six degrees of freedom (three in translation and three in rotation) that a volume has in the 3D space [96]. The intensity-based method makes a registration using pixel-wise information of 2D images. A merit function is used to determine the similarity of the images quantitatively [5]. Consequently, the registration is achieved by finding the maximum of the merit function by varying the pose given to the DRR [97]. It is worth to mention that intensity-based registration is the most reported method for 2D/3D registration [58] [98].

Although there is not a consensus about the best method for 2D to 3D registration, common patterns in the literature show that best results are obtained from intensity-based methods when working with imaging modalities from CT and X-rays images [58] [5] [99] [91]. As CT-scans and X-rays are based on the same natural phenomenon, creating a DRR from a 3D model made from slides of X-rays leads to similar results. On the other hand, creating a DRR from an MR generates a different outcome than an X-ray, so the results would not be optimal [57].

For this work, the 3D-volume data sets were created from CT-scans and the 2D images from X-rays. Therefore, intensity-based methods will be used as the base for the registration procedure. In the following chapters, more details regarding 2D/3D registration using intensity-based methods are discussed, i.e., the working principle of the DRR, type of merit functions, optimizers, and the registration algorithm.

4.7. Intensity-Based Methods Based on Digitally Reconstructed Radiograph (DRR)

This work will focus on a 2D/3D registration using intensity-based methods, registering two types of modalities, 2D and 3D. The 2D modality is an X-ray image, and the 3D modality is a CT-scan. The registration is an optimization process that compares the CT-scan with a set point established by the X-ray image. The comparison is executed in the $\mathbb{R}^2$ space by projecting the CT-scan into $\mathbb{R}^2$ with a digitally reconstructed radiograph (DRR) procedure, which is explained in section 4.7.1. The projection of the DRR depends on the six DoF available in $\mathbb{R}^3$. The DRR image and the X-ray are compared with a similarity measurement function, explained in section 4.7.2. The union of the two modalities, the six DoF to generate a DRR image, and the image similarity function define the 2D/3D registration cost function. To create the gradient of the cost function, the X-ray is compared with DRR images created with delta changes in every DoF. Then the gradient is evaluated by an optimization function that gives a new pose for the DRR. More development on the optimizers is given in section 4.7.3. From this basic introduction, it can be noticed that the selected registration procedure is an iterative process, elaborated in more detail in section 4.7.4. Most of the numerical optimization algorithms require an initial guess to find the optimal result. The algorithm for 2D/3D registration is not an exemption to this rule; on the contrary, initialization is a fundamental step to achieve a successful result. In sections 5.7 and 7.2, there are more details regarding the initial pose selection.

4.7.1. Digitally Reconstructed Radiograph (DRR) Module

The digitally reconstructed radiograph (DRR) module is one of the fundamentals pieces that form the 2D/3D registration. It helps to render images that are compared with a baseline X-ray image. One way to understand the working principle of the DRR module is by making an analogy with the C-arm. As explained in section 4.5, a C-arm consists of an X-ray source at one end of its C-shape structure and an X-ray detector at the other end. The radiation pierces a patient lying in the C-arm, and the detector forms an X-ray image by capturing the attenuated radiation due to the body segment. Now imagining a C-arm in the digital world, but instead of a patient, there is a digital 3D volume, e.g., a CT-scan. The rays go through the CT-scan and generate an X-ray image out of a 3D volume. Thus, the resulted X-ray image is instead called a digitally reconstructed radiograph.

In computer vision, creating a 2D image from a 3D object is called rendering [100]. One of the most usual rendering methods is volume ray-casting. It offers high-quality rendering as no simplifications are involved [101] [99]. In brief, a virtual ray is created from each pixel of the resulted image to the ray source. A line integral is calculated for each of the rays while the ray goes through the 3D object [102]. In the DRR module, the ray source is punctual and is located in a known position with respect to the 3D object. Then the resulted image is created with a perspective projection. Following this idea, the DRR model follows the same principle as the pinhole camera explained in section 4.5.3, so the nature of the DRR is equivalent to that of the C-arm. It can be noticed that the reference frame of the DRR module and the 3D object belongs to a three-dimensional space, so the description of the object with respect to the DRR frame requires six DoF: three positions and three orientations. Consequently, a DRR image can be generated by defining three translations, $[t_x, t_y, t_z]$, and three rotations $[r_x, r_y, r_z]$. This six DoF are gathered in the vector $\vec{\boldsymbol{p}} \epsilon \mathbb{R}^6$ described in in equation (*47*).

$$\vec{\boldsymbol{p}} = [t_x, t_y, t_z, r_x, r_y, r_z]^T \tag{47}$$

Figure 4-21 shows how the input of the DRR, i.e., a CT-scan of a spine model, is projected into a 2D image given the impression of an X-ray image.

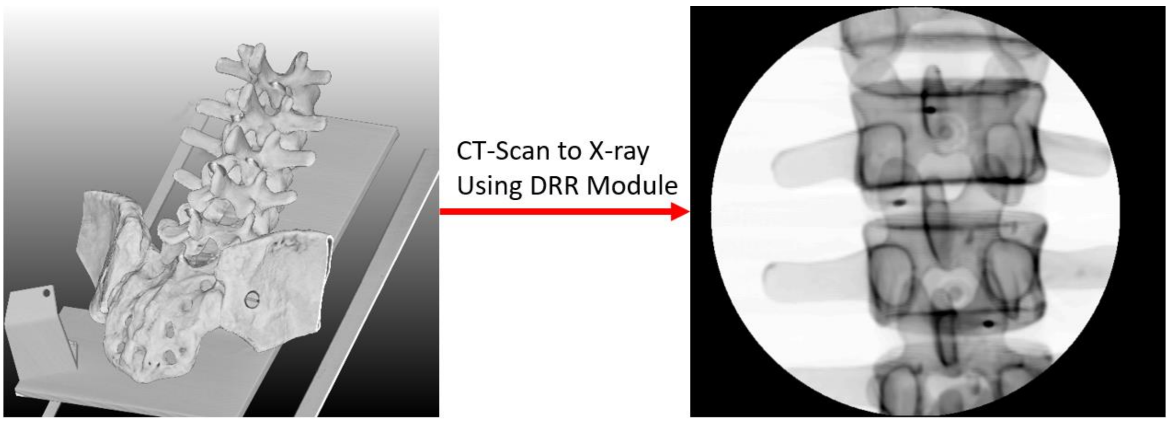

Figure 4-21. CT-scan to X-ray using the DRR module

As a point of comparison, an X-ray of the same spine model taken with the C-arm is shown in Figure 4-22.

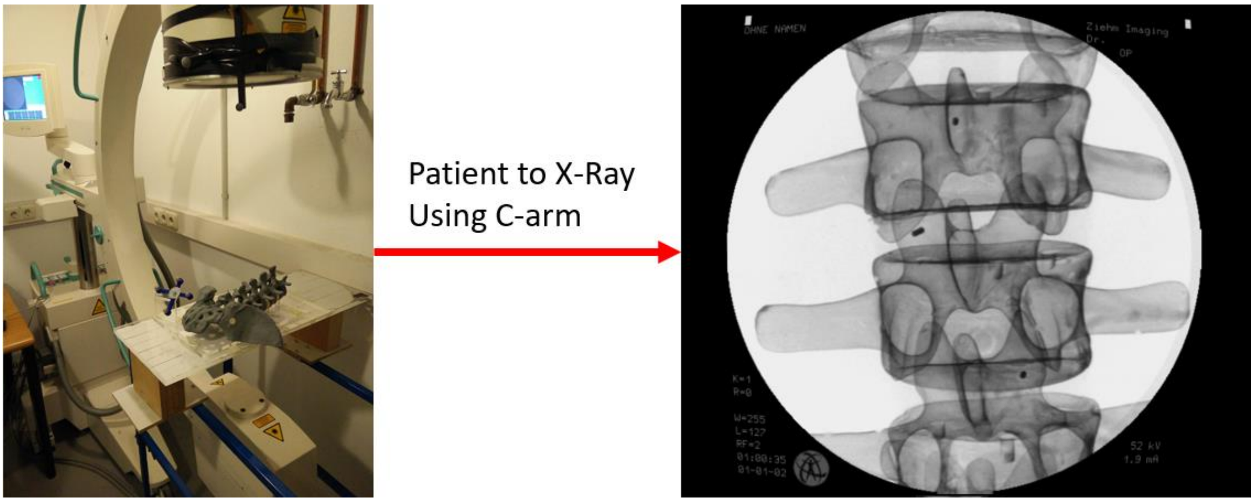

Figure 4-22. Patient to X-ray using the C-arm

It can be noticed that the visualization of both images shows similarly the bone structure, i.e., showing darker spots where the bone density is higher, lighter ones when the bone density is lower. As a formality, the mathematical function representing the creation of a digitally reconstructed radiograph of a CT-scan at a specific pose $\vec{\boldsymbol{p}} \epsilon \mathbb{R}^6$ is denoted as:

$$f_{DRR}(\boldsymbol{V}_{CTscan}\ ,\vec{\boldsymbol{p}}) \quad (48)$$

The DRR process has its reference frame {DRR}, and it actuates over the DICOM modality, which has a reference frame {DICOM}. Then $\vec{\boldsymbol{p}}$ expresses the transformation ${}^{DRR}\boldsymbol{T}_{DICOM}$. One desired feature of the DRR module is to match the DICOM reference frame (see Figure 4-20) when there is no rotation and translation. Additionally, the ray source is located in the negative Y-axis, and the rays move along the positive Y-axis until reaching the rendering plane. The origin of {DRR} lies in the middle point between the rendering plane and the ray source. The distance from the source to the rendering (detector) plane is known as the focal length (f) in the pinhole camera model. In the DRR module, f is a user-defined parameter. It influences the DRR image rendering as if the source-detector distance could be adjustable in a C-arm device. On the rendering plane, the X-axis expresses the first element and the Z-axis the second element of the 2D coordinate representing a pixel position within the rendered image. The previous considerations are depicted in Figure 4-23.

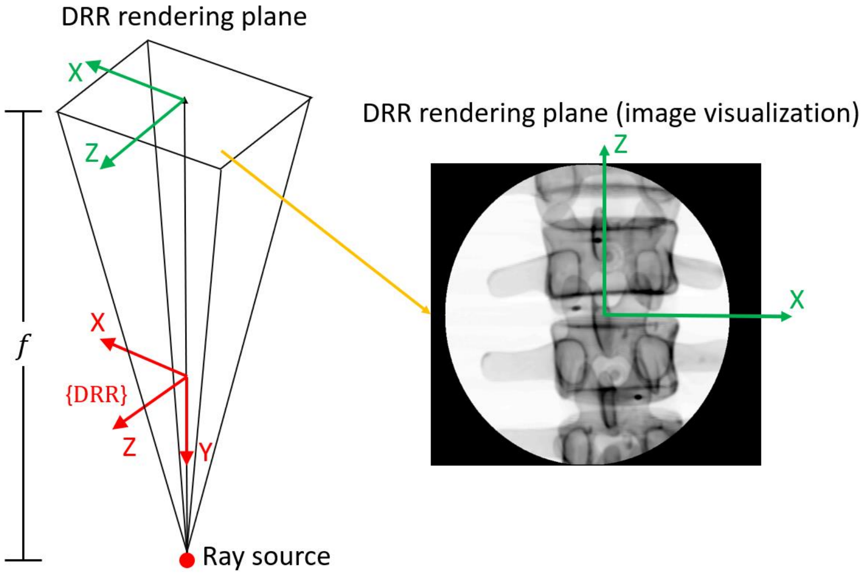

Figure 4-23. DRR reference frames representation

Siddon Algorithm

For radiological applications, the first ray casting algorithm was called the *Siddon algorithm* [103]. The algorithm performs the volume ray casting by establishing the equation for the radiological path as:

$$d = \sum_i \sum_j \sum_k l(i,j,k)\rho(i,j,k) \tag{49}$$

Where $\rho(i,j,k)$ is the voxel intensity value and $l(i,j,k)$ the ray length contained by that voxel.

Equation (*49*) is a weighted sum that substitutes the line integral in discrete space, where the weights in the sum are the lengths of the ray segments inside each voxel.

Considering voxels as the intersection volumes of orthogonal sets of equally spaced parallel planes, the Siddon algorithm calculates the intersection of a ray with the planes, as Figure 4-24 shows. The algorithm represents the position of every single ray with three parametric equations, one for each axis:

$$\begin{aligned} X(\alpha) &= X_A + \alpha(X_B - X_A) \\ Y(\alpha) &= Y_A + \alpha(Y_B - Y_A) \\ Z(\alpha) &= Z_A + \alpha(Z_B - Z_A) \end{aligned} \tag{50}$$

Where the parameter α is zero at point A and one at point B [103].

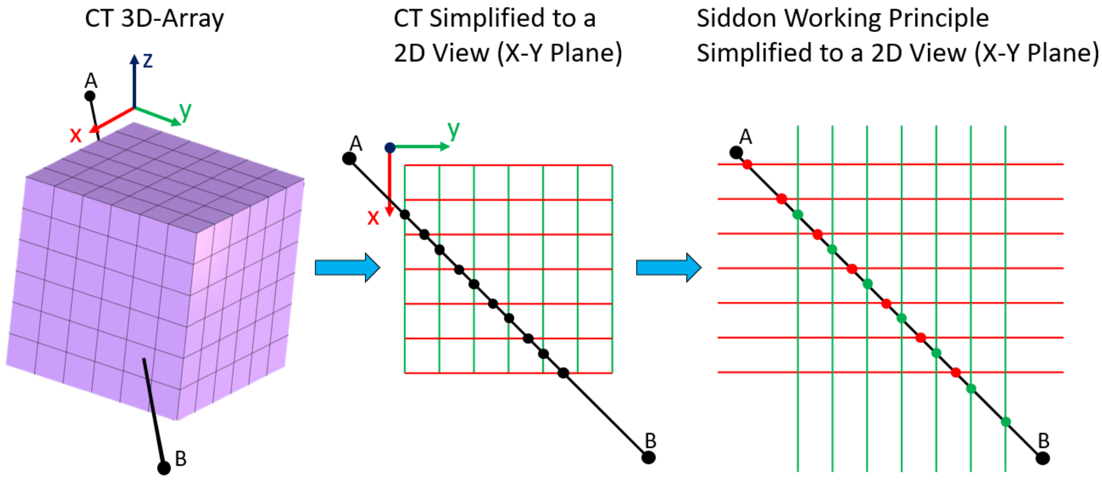

Figure 4-24. Working principle of the Siddon algorithm. Intersections with vertical planes marked with green dots. Intersections with horizontal planes marked with red dots

The points α_{min} and α_{max}, where the ray goes in and out of the volume, are shown in Figure 4-25. The Siddon algorithm determines the ray intersection values with each set of planes and merges the three resultant sets of parametric values into one set. Thus, the length of the ray $l(i,j,k)$ contained in every voxel can be determined knowing the consecutive intersects. The corresponding intensity value $\rho(i,j,k)$ is obtained from the CT voxel, and the radiological path can be calculated using equation (*49*).

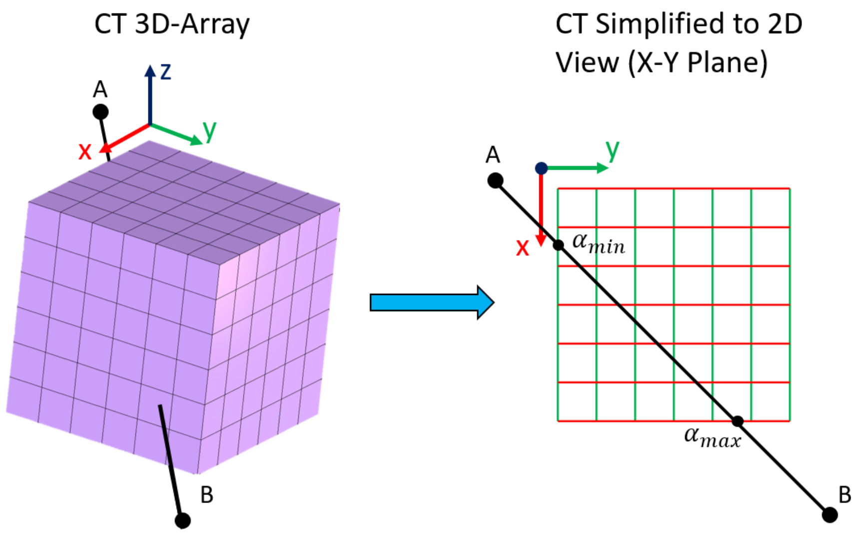

Figure 4-25. Visualization of the points α_{min} and α_{max} in the Siddon algorithm

Siddon-Jacobs Algorithm

There is a faster version of the previous algorithm, the Siddon-Jacobs algorithm. It follows the same flow as the Siddon algorithm, but the Siddon-Jacobs algorithm does not calculate the ray intersection with every single plane. Instead, it calculates only the value of the first intersection point of the ray with the planes after the ray entered the volume. Then the length of the ray between planes is calculated and decomposed in the length of each axis. Figure 4-26 shows the lengths as α_{xu} and α_{yu}. It can be noticed that those lengths remain constant for the entire ray path. Knowing these lengths and the distance between planes, d_{x} and d_{y} , the intersects are calculated. When the intersects are found, the same procedure used for the Siddon algorithm is used. The authors claim to achieve a speedup of 7.5 for calculating the radiological path compared with the Siddon Algorithm [104]. A 2D representation of the Siddon-Jacobs principle is shown in Figure 4-26.

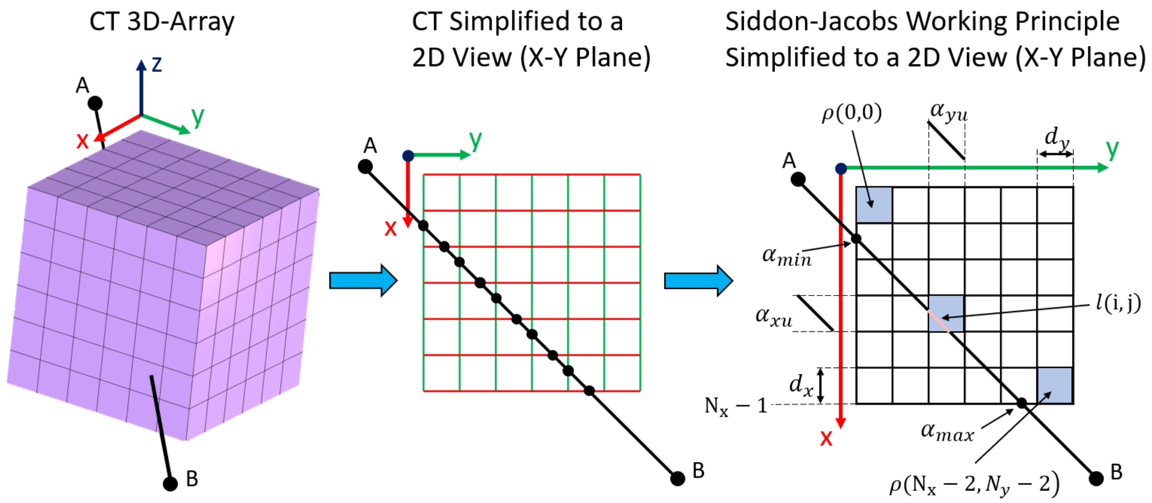

Figure 4-26. Working principle of Siddon-Jacobs algorithm

4.7.2. Image Similarity Measurements (Merit) Functions

An X-ray from the patient and a DRR image projected from the pre-operative CT-data are used to guide the algorithm to find the registration. The typical approach uses a merit function with the two images as inputs and a numerical value as the output [105]. Two images can be compared qualitatively by expressing how similar they look, but the result could not be used as the input of a subsequent mathematical procedure. Then it is required to use a quantitative measurement that computes a number depending on the similarity of two images. In this sense, an image similarity measurement, also called merit function, assigns a value to the similarity of two images.

It is possible to compare images considering only the intensities contained in the image pixels. These functions are called intensity-only similarity measurements. Other measurements use the pixel values and their spatial distribution among the images, known as spatial-information similarity measurements. Further measurements compute the numerical distribution of the pixel values, called histogram-based similarity measurements.

Now some concepts will be introduced to define more precisely the similarity measurements:

An *image* is defined as a bi-dimensional discrete function, $\boldsymbol{I}(x, y)$ [106]. The independent variables x and y represent the columns and rows of the image, respectively. The intersection of a row and a column represents a *pixel*, which has a discrete intensity value. The domain of the image is defined by $u \in \mathbb{Z}$, the *number of columns*, and $v \in \mathbb{Z}$, the *number of rows*. The range of the function is limited by $Q \in \mathbb{Z}$, the *maximum intensity level*. In this work, only grayscale images are considered; therefore, only one channel is used. Q is the largest number that the binary representation allows, i.e., $2^q - 1$, where $q \in \mathbb{Z}$ represents the bit depth. Mathematically, the previous definitions are described by equation (*51*).

$$Q = 2^q - 1$$
$$\{\boldsymbol{I}(x,y), x, y, u, v \in \mathbb{Z} \,;\, (x,y) \in \mathbb{Z}^2 |\ 0 \le \boldsymbol{I}(x,y) \le Q \ ;\ 0 \le x \le u-1 \ ;\ 0 \le y \le v-1\} \tag{51}$$

The domain of the image will be further defined as the set T. The image boundaries are still valid if x and y belong to the set $\mathbf{T}$ as equation (*52*).

$$(x,y) \in \boldsymbol{T} \ if \ (0 \le x \le u-1) \wedge (0 \le y \le v-1) \tag{52}$$

The *number of pixels* in an image is defined as $N \in \mathbb{Z}$, which is the product of u and v.

$$N = u \cdot v \tag{53}$$

Mathematically, the operation of finding the similarity between two images regardless of the used function will be denoted as:

$$f_{similarity}(\boldsymbol{I}_1, \boldsymbol{I}_2) \tag{54}$$

4.7.2.1. Intensity-Only Similarity Measurements

This first set of merit functions computes the similarity value using the intensity values pixel-wise. That means, they do not consider the intensity values of neighboring pixels or the spatial distribution of the intensity values. Because of their definition, it is necessary that each pixel of one image has a unique correspondence in the other image. It means both images must have the same size.

Sum of Squared Differences (SSD)

SSD is the simplest of the merit functions. It consists of the pixel-wise difference of both images followed by a pixel-wise square operation. After that, the mean value of the resulting image is calculated as equation (*55*) shows. If the two images only differ by Gaussian noise, SSD is the most optimal measurement [107]. SSD is very sensitive to small images with large differences in grayscale values, i.e., the images have different contrast [108].

$$SSD(\boldsymbol{I}_1, \boldsymbol{I}_2) = \frac{1}{N} \sum_{(x,y) \epsilon \boldsymbol{T}} \left(\boldsymbol{I}_1(x,y) - \boldsymbol{I}_2(x,y) \right)^2 \tag{55}$$

Sum of Absolute Differences (SAD)

This merit function is also known as ℓ_1 norm or Manhattan distance. It consists of the pixel-wise difference of both images followed by an absolute value operation. After that, the mean value of the resulting image is calculated as equation (*56*) shows. SAD improves the SSD sensitivity problem for small images and images with a large difference in contrast [109]. This merit function is widely used in video encoding due to its simplicity and accurate matching results [110].

$$SAD(\boldsymbol{I}_1, \boldsymbol{I}_2) = \frac{1}{N} \sum_{(x,y) \epsilon \boldsymbol{T}} |\boldsymbol{I}_1(x,y) - \boldsymbol{I}_2(x,y)| \tag{56}$$

Normalized Cross-Correlation (NCC)

This merit function is used to evaluate the linear dependence of two variables. For the case of two images, it is used to find the linear dependency of the pixels on the first image, with respect to the second [111]. It is also known in statistics as the Pearson product-moment correlation, or simply Pearson correlation coefficient, and is shown in equation (*57*). It has the advantage of being insensitive to differences in the noise levels of the input images [112].

$$NCC(\boldsymbol{I}_1, \boldsymbol{I}_2) = \frac{1}{\sigma_1 \sigma_2 N} \sum_{(x,y)\epsilon \boldsymbol{T}} (\boldsymbol{I}_1(x,y) - \overline{\boldsymbol{I}}_1)(\boldsymbol{I}_2(x,y) - \overline{\boldsymbol{I}}_2) \tag{57}$$

Where $\overline{\boldsymbol{I}}_k$ is the average of the image *k*, and $\sigma_k$is the standard deviation of the image *k*.

Normalized Absolute Cross-Correlation (NACC)

This metric function based on the NCC is proved to be a more accurate estimator than NCC [113]. Using NCC, some inversely-related values are compensated, which is not the case where using NACC. The equation for computing NACC is presented in (*58*).

$$NACC(\boldsymbol{I}_1, \boldsymbol{I}_2) = \frac{1}{\sigma_1 \sigma_2 N} \sum_{(x,y)\epsilon \boldsymbol{T}} |(\boldsymbol{I}_1(x,y) - \overline{\boldsymbol{I}}_1)(\boldsymbol{I}_2(x,y) - \overline{\boldsymbol{I}}_2)| \tag{58}$$

4.7.2.2. Spatial-Information Similarity Measurements

The following merit functions still use the pixel intensity but also consider the pixel spatial distribution within the image. The spatial information is obtained by considering the surrounding pixels, as it is the case with pattern intensity, sum of local normalized cross-correlation and variance-weighted sum of local normalized correlation, or with a pre-operation that uses the neighboring pixels, e.g., gradient operation, in the case of gradient correlation and gradient difference.

For these merit functions, some further definitions will be introduced.

Define the image difference of $\boldsymbol{I}_1(x,y)$ and $\boldsymbol{I}_2(x,y)$, as the pixel-wise difference of $\boldsymbol{I}_1$ and $\boldsymbol{I}_2$ multiplied by a scaling factor $s \in \mathbb{Z}$, which balances the contrast between the images. The difference operation is written in equation (*59*).

$$\boldsymbol{I}_{diff}(x,y) = \boldsymbol{I}_1(x,y) - s \cdot \boldsymbol{I}_2(x,y) \tag{59}$$

Pattern Intensity (PI)

PI works directly with the image difference created by the X-ray and the DRR image. Each pixel of the image difference is evaluated along with its surrounding pixels within a circle of radius r. PI defines that a pixel belongs to a structure if the intensity of the evaluated pixel and their surroundings is high. From another perspective, if two images are identical, there would not be any structure [114]. PI evaluates the “structuredness” by assigning small values where there are significant changes in intensities and high values where the intensity values do not change too much [115]. A small PI measurement indicates, therefore, that the images are not similar, while a high PI value indicates that both images are very similar. PI is defined by the equations (*60*), and (*61*). The output of PI increases with the similarity of both images and with the size of the evaluated images. It is important to keep the same size of the images for the entire registration procedure as the results must be used by comparing with other PI values. Define the circle of neighbored pixels as:

$$d^2 = (x - x_v)^2 + (y - y_w)^2 < r^2 \tag{60}$$

where (x_v, y_w) are pixels belonging to the image domain and within the circle center in (x, y) with radius r. d is the distance of the coordinates (x_v, y_w) and the circle center (x, y).

Keeping the circle of neighboring pixels in mind, the pattern intensity can be computed as:

$$PI(\boldsymbol{I}_1, \boldsymbol{I}_2, \sigma_s) = \sum_{(x,y)\epsilon \boldsymbol{T}} \sum_{d^2 \leq r^2} \frac{{\sigma_s}^2}{{\sigma_s}^2 + \left(\boldsymbol{I}_{diff}(x,y) - \boldsymbol{I}_{diff}(v,w)\right)^2} \tag{61}$$

where σ_s is a sensitivity parameter defining the intensity level to be considered a structure.

The values used for σ_s and r were 10 and 3 respectively, which are commonly used in literature [114] [115] [116].

Gradient Correlation (GC)

GC starts by differentiating the two input images, $\boldsymbol{I}_1(x, y)$, and $\boldsymbol{I}_2(x, y)$. The derivatives are calculated using isotropic 3x3 image gradient operators, known as Sobel kernels [117]. For each image, the vertical and horizontal gradients are computed, i.e., $\frac{\partial \boldsymbol{I}_1}{\partial x}, \frac{\partial \boldsymbol{I}_1}{\partial y}, \frac{\partial \boldsymbol{I}_2}{\partial x}$ and $\frac{\partial \boldsymbol{I}_2}{\partial y}$. The vertical gradients of image 1 and 2, i.e., $\frac{\partial \boldsymbol{I}_1}{\partial y}$ and $\frac{\partial \boldsymbol{I}_2}{\partial y}$, are used to compute the normalized cross-correlation using the equation (*57*) and the result is denoted as $\mathrm{NCC}_{\frac{\partial}{\partial y}}$. Also, the correlation is calculated for the gradients of the columns, $\frac{\partial \boldsymbol{I}_1}{\partial x}$ and $\frac{\partial \boldsymbol{I}_2}{\partial x}$ and is named as $\mathrm{NCC}_{\frac{\partial}{\partial x}}$. At the end, both results are averaged, creating the GC value [118]. The equation (*62*) shows the expression for the gradient correlation.

$$GC(\boldsymbol{I}_1, \boldsymbol{I}_2) = \frac{\left(NCC_{\frac{\partial}{\partial x}} + NCC_{\frac{\partial}{\partial y}}\right)}{2} \tag{62}$$

It is worth to note that the application of vertical and horizontal Sobel kernels incorporates in the measurement the knowledge of neighbor pixels. Additionally, the application of Sobel kernels focused on well-defined edges. It implies that soft tissue is not considered in the correlation measurement, and bone structures are the main contributors to the measurement [57].

Gradient Difference (GD)

As in the case of GC, GD also starts by differentiating $\boldsymbol{I}_1(x, y)$, and $\boldsymbol{I}_2(x, y)$. The difference operator in (*59*) is applied over $\frac{\partial \boldsymbol{I}_1}{\partial y}$ and $\frac{\partial \boldsymbol{I}_2}{\partial y}$., and the result is called $\boldsymbol{I}_{diffV}(x, y)$. Likewise, the same operation is applied using $\frac{\partial \boldsymbol{I}_1}{\partial x}$ and $\frac{\partial \boldsymbol{I}_2}{\partial x}$ in equation (*59*). The result is named $\boldsymbol{I}_{diffH}(x, y)$. These operations can be seen in equation (*63*). The same form $\frac{a}{a+x^2}$ as in Pattern Intensity, see equation (*61*), is used in both image differences. Then both results are added, founding the GD value [118]. The operation can be seen in equation (*64*).

$$\begin{aligned} \boldsymbol{I}_{diffH}(x, y) &= \frac{\partial \boldsymbol{I}_1}{\partial x} - s \cdot \frac{\partial \boldsymbol{I}_2}{\partial x} \\ \boldsymbol{I}_{diffV}(x, y) &= \frac{\partial \boldsymbol{I}_1}{\partial y} - s \cdot \frac{\partial \boldsymbol{I}_2}{\partial y} \end{aligned} \tag{63}$$

$$GD(\boldsymbol{I}_1, \boldsymbol{I}_2, s) = \sum_{(x,y)\epsilon \boldsymbol{T}} \frac{A_v}{A_v + \left(\boldsymbol{I}_{diffV}(x, y)\right)^2} + \sum_{(x,y)\epsilon \boldsymbol{T}} \frac{A_h}{A_h + \left(\boldsymbol{I}_{diffH}(x, y)\right)^2} \tag{64}$$

where A_v and A_h are sensitivity parameters defining the intensity level to be considered a structure.

This measurement penalizes the formation of structures as well as PI, but in the case of GD, it provides sensitivity for thin edges as it works with the difference of gradient images [57]. The values used for A_v and A_h are set to the variance of the corresponding gradients of the X-ray image [57] [118] [99].

Sum of Local Normalized Cross-Correlation (SLNCC)

Although NCC is insensitive to differences in the noise levels of the input images [112], the C-arm adds spatially intensity distortions due to non-uniformities in the image intensifier, leading to bias errors when using NCC [27]. A way to overcome this phenomenon is to evaluate NCC in sets of small neighborhoods that together cover the totality of the images. It is important to mention that each set evaluates the same pixels in image 1 and image 2. The result of the NCC of every set is added and divided by the size of the number of pixels in the set [119] [99]. See equation (*65*). This process is called *sum of local normalized cross-correlation*.

Define a sub-image of the image n of size $p \times q$ starting in the position i, j as ${}_{pq}^{ij}\boldsymbol{I}_n$.

$$SLNCC(\boldsymbol{I}_1, \boldsymbol{I}_2, p, q, d) = \frac{1}{p \cdot q} \sum_{i=1,(1+d),(1+2d),\dots}^{u-p} \sum_{j=1,(1+d),(1+2d),\dots}^{v-q} NCC\left({}_{pq}^{ij}\boldsymbol{I}_1, {}_{pq}^{ij}\boldsymbol{I}_2\right) \tag{65}$$

where d is the stride of the set displacement of pixel that moves each time.

The measurement range of this function goes from 1, when the images are correlated, to -1, which indicates anticorrelation. The selected values for p and q are 7 and 1 for d, based on the reported literature [119] [99]. The pixels in a set may have the same intensity value, which makes the standard deviation zero; therefore, the NCC is undetermined as equation (*57*) shows. To avoid this problem, Gaussian-noise of a small magnitude is added to the DRR image.

Variance-Weighted Sum of Local Normalized Correlation (VWSLNC)

Previously with SLNCC, it was shown how the set of small neighboring areas is evaluated with NCC and then added. However, all sets are weighted equally. In order to give more importance to areas with higher information, the NCC value of each region is multiplied by the variance within that region of one of the images used to compute the NCC. After that, the NCC weighted values are added, and the result is divided by the summation of all the variances [119]. See equation (*66*).

Define the standard deviation of the sub-image ${}^{ij}_{pq}\boldsymbol{I}_n$.as ${}^{ij}_{pq}\sigma_n$

$$VWSLNC(\boldsymbol{I}_1, \boldsymbol{I}_2, p, d) = \frac{\sum_{i=1,(1+d),(1+2d),\ldots}^{u-p} \sum_{j=1,(1+d),(1+2d),\ldots}^{v-q} \left[{}^{ij}_{pq}\sigma_1 \cdot NCC\left({}^{ij}_{pp}\boldsymbol{I}_1, {}^{ij}_{pp}\boldsymbol{I}_2\right)\right]}{\sum_{i=1,(1+d),(1+2d),\ldots}^{u-p} \sum_{j=1,(1+d),(1+2d),\ldots}^{v-q} {}^{ij}_{pq}\sigma_1} \tag{66}$$

The measurement range of this function goes from 1, when the images are correlated, to -1, which indicates anticorrelation. The selected values for p,q and d are the same as is in the case of SLNCC.

4.7.2.3. Histogram-Based Similarity Measurements

The merit functions described in this chapter do not operate on the images. Instead, they are applied to the image histograms [120]. The pixels intensities are still used to create the histogram, but the spatial information of the pixels is replaced by the distribution of the intensities within the image. As a consequence, these measurements have the advantage that images with different sizes can now be compared, but they have the disadvantage of lacking spatial information.

Before entering in detail to these similarity measurements, some further definitions must be introduced. Within this section, the process is assumed to be ergodic. Although these measurements are called histogram-based, they make use of the empirical estimate of the probability distribution of the image by means of a normalized frequency of occurrence. The probability distribution is approximated, dividing the frequency of each bin of the histogram by the number of pixels in the image. Let $\boldsymbol{\Omega}$ be the sample space of the intensities of an image with maximum intensity Q as:

$$\boldsymbol{\Omega} = \{0, 1, \ldots, Q\} \tag{67}$$

Now let $\mathbf{A} \subseteq \boldsymbol{\Omega}$ be the sample space of the image $\boldsymbol{I}_1$, and the intensity $a \in \mathbf{A}$ an event of the sample space $\mathbf{A}$. The probability of finding a pixel of intensity a in $\boldsymbol{I}_1$ is denoted as:

$$\mathbb{P}_1(a) \tag{68}$$

Another measurement used is the joint probability distribution, which is estimated from the joint histogram divided by the product of pixels in two independent images, image 1 and image 2. let $\mathbf{B} \subseteq \mathbf{\Omega}$ be the sample space of the image $\boldsymbol{I}_2$, and the intensity $b \in \mathbf{B}$ an event of the sample space $\mathbf{B}$. The joint histogram is two dimensional and is done by "plotting the intensity a of each voxel in image 1 against the intensity b of each voxel in image 2" [121]. The joint probability distribution of two images can be read as the probability to find a pixel of intensity a in the image 1 and a pixel of intensity b in image 2, and it is denoted as:

$$\mathbb{P}(a, b) \tag{69}$$

Mutual Information (MI)

MI acts over the discrete probability distribution of the images. Mutual information was initially applied to stochastics methods as a way to express the increase of information of a discrete random variable based on the knowledge of a second random variable [122]. MI measures the distance between the joint probability distribution function using equation (*69*) and the joint probability distribution function assuming independent random variables using (*68*). In other words, it measures the dependence of two images [123]. MI is defined in equation (*70*) as:

$$MI(\boldsymbol{I}_1, \boldsymbol{I}_2) = \sum_{a,b} \mathbb{P}(a, b) \cdot log \frac{\mathbb{P}(a, b)}{\mathbb{P}_1(a)\, \mathbb{P}_2(b)} \tag{70}$$

where $\mathbb{P}_1(a)$ is the probability of the intensity a in the image 1, and $\mathbb{P}_2(b)$ is the probability of the intensity b in the image 2.

Mutual Information gets close to zero the more independent the both probability distributions are, i.e., when the images are entirely unrelated. However, the upper border is unknown. It is only known that the Mutual information is less than the Shannon's entropy of any of both images [124].

Normalized Mutual Information (NMI)

As the upper range of MI is unknown, NMI is an attempt to normalize MI, so that its range is between 0 to 1 [125]. The mathematical formulation for calculating NMI can be seen in equation (*71*).

$$NMI(\boldsymbol{I}_1, \boldsymbol{I}_2) = \frac{2 \cdot MI(\boldsymbol{I}_1, \boldsymbol{I}_2)}{-\sum_a \mathbb{P}_1(a) \cdot log\big(\mathbb{P}_1(a)\big) - \sum_b \mathbb{P}_2(b) \cdot log\big(\mathbb{P}_2(b)\big)} \tag{71}$$

Correlation Ratio (CR)

CR was designed as an extension of normalized cross-correlation, which was explained in section 4.7.2.1. Instead of measuring the linear dependence of variables as NCC, CR measures the functional dependence of two discrete random variables [126]. An image can be depicted as a discrete random value by using the estimate of the underlying probability distribution function obtained from its histogram, as outlined before. The more functional dependent is an image with respect to the other one, the more similar they are. It can be seen in equation (*72*), that the two variables do not have the same purpose in the functional relationship. It makes CR asymmetrical by nature, which means $CR(\boldsymbol{I}_1, \boldsymbol{I}_2) \neq CR(\boldsymbol{I}_2, \boldsymbol{I}_1)$ [127].

$$\sigma^2 = \left(\sum_b b^2 \cdot \mathbb{P}_2(b)\right) - \left(\sum_b b \cdot \mathbb{P}_2(b)\right)^2$$
$$\sigma_a^2 = \left(\frac{1}{\mathbb{P}_1(a)} \sum_b b^2 \cdot \mathbb{P}(a,b)\right) - \left(\frac{1}{\mathbb{P}_1(a)} \sum_b b \cdot \mathbb{P}(a,b)\right)^2 \quad (72)$$
$$CR(\boldsymbol{I}_1, \boldsymbol{I}_2) = 1 - \frac{1}{\sigma^2} \sum_a \sigma_a^2 \cdot \mathbb{P}_1(a)$$

The range of CR goes from 0, when there is no functional dependence between images, and 1, when the dependence is at highest, meaning both images are purely deterministically dependent [127]. Care must be taken when selecting which image is the reference image, $\boldsymbol{I}_1$, and which the floating, $\boldsymbol{I}_2$, because the results vary with such selection. During the registration process, the X-ray image remains constant while several DRR images are generated. Therefore, the X-ray is selected to be the reference image in CR, and the DRR image is chosen as the floating image.

4.7.3. Optimization Algorithms in 2D/3D Registration

Up to this point, it is clear that an X-ray image and a DRR image can be compared using a merit function as the ones explained in section 4.7.2. Bearing that in mind, an optimization process is used to find the minimum value of a cost function depending on the input pose. The cost function is built by evaluating the merit function with the DRR image and the X-ray image. In other words, the cost function is the pipeline starting with the creation of the DRR image from the CT-Volume, defined in equation (*48*), using the parameters $\vec{\boldsymbol{p}} \epsilon \mathbb{R}^6$ as in equation (*47*), and comparing the result, i.e., DRR image with the X-ray using a merit function described in equation (*54*). Then, the cost function depends on six independent variables, $\vec{\boldsymbol{p}}$, an X-ray image, $\boldsymbol{I}_{Xray}$, and a 3D CT-scan volume, $\boldsymbol{V}_{CTscan}$.

$$f_c\left(\boldsymbol{I}_{Xray}, \boldsymbol{V}_{CTscan}, \vec{\boldsymbol{p}}\right) = f_{similarity}\left(\boldsymbol{I}_{Xray}, f_{DRR}(\boldsymbol{V}_{CTscan}, \vec{\boldsymbol{p}})\right) \quad (73)$$

The DRR algorithm includes non-linear operations, and all merit functions are non-linear. That means the function to optimize is non-linear. Therefore, a non-linear optimizer is suitable for this work. Due to the nature of the problem, there is no information beforehand about the possible range of the DRR pose for a specific 3D dataset, making it difficult to constrain the optimization [128]. In other words, this is an unbounded optimization problem, which will be approached with a non-linear optimizer. As a final consideration, it has been reported that non-gradient-based optimizers, as downhill simplex, take several hundreds of function evaluations before reaching the convergence point [119] [129] [130]. One function evaluation takes in the order of tens of milliseconds; therefore, the registration would take tens of seconds. One of the purposes of this work is to make a registration in a few seconds; consequently, the evaluated optimizers are unbounded non-linear gradient-based optimizers. Gauss-Newton and Levenberg-Marquardt are an excellent option for optimizers when the cost function is the sum of non-linear squared functions [131], which is the case only in sum of square difference (SSD). For that reason, these two optimizers were not considered in this work as an option for the registration process.

Before introducing the specific optimizers considered in this work, some definitions must be presented.

The cost function $f_c(\boldsymbol{I}_{Xray}, \boldsymbol{V}_{CTscan}, \vec{\boldsymbol{p}})$ is assumed to be differentiable $\forall \vec{\boldsymbol{p}} \in \mathbb{R}^6$.

Defining the gradient of the cost function with respect to $\vec{\boldsymbol{p}}$, $\nabla_{\vec{\boldsymbol{p}}} f_c(\boldsymbol{I}_{Xray}, \boldsymbol{V}_{CTscan}, \vec{\boldsymbol{p}})$. The gradient with respect to $\vec{\boldsymbol{p}}$ is also an $\mathbb{R}^6$ vector made of the partial derivative of the cost function as the equation (*74*) shows.

$$\vec{\nabla}_{\boldsymbol{p}} f_c(\boldsymbol{I}_{Xray}, \boldsymbol{V}_{CTscan}, \vec{\boldsymbol{p}}) = \begin{bmatrix} \frac{\partial f_c(\boldsymbol{I}_{Xray}, \boldsymbol{V}_{CTscan}, \vec{\boldsymbol{p}})}{\partial t_x} \\ \frac{\partial f_c(\boldsymbol{I}_{Xray}, \boldsymbol{V}_{CTscan}, \vec{\boldsymbol{p}})}{\partial t_y} \\ \frac{\partial f_c(\boldsymbol{I}_{Xray}, \boldsymbol{V}_{CTscan}, \vec{\boldsymbol{p}})}{\partial t_z} \\ \frac{\partial f_c(\boldsymbol{I}_{Xray}, \boldsymbol{V}_{CTscan}, \vec{\boldsymbol{p}})}{\partial r_x} \\ \frac{\partial f_c(\boldsymbol{I}_{Xray}, \boldsymbol{V}_{CTscan}, \vec{\boldsymbol{p}})}{\partial r_y} \\ \frac{\partial f_c(\boldsymbol{I}_{Xray}, \boldsymbol{V}_{CTscan}, \vec{\boldsymbol{p}})}{\partial r_z} \end{bmatrix} \tag{74}$$

The optimization process, to obtain the parameters $\vec{\boldsymbol{p}} \epsilon \mathbb{R}^6$ that make the cost function minimum, is defined mathematically in equation (*75*),

$$\underset{\vec{\boldsymbol{p}} \epsilon \mathbb{R}^6}{argmin} \; f_c(\boldsymbol{I}_{Xray}, \boldsymbol{V}_{CTscan}, \vec{\boldsymbol{p}}) \tag{75}$$

using $\vec{\boldsymbol{p}}_0$ as the process initial pose.

It is worth noting that, although the results of the optimization are affected by $\boldsymbol{I}_{Xray}, \boldsymbol{V}_{CTscan}$ these 2D and 3D imaging remain constant throughout the optimization process. That means, once the optimization process starts, only $\vec{\boldsymbol{p}}$ changes, and therefore only the gradient with respect to $\vec{\boldsymbol{p}}$ is defined. For the following definition of the optimizers, the cost function, f_c, will be considered to depend only on the variable $\vec{\boldsymbol{p}} \epsilon \mathbb{R}^6$, i.e., $f_c(\vec{\boldsymbol{p}})$.

Best Neighbors (BN)

BN is a non-gradient optimizer best known in the literature as Hill-Climbing. It has been tested, despite being mentioned in the preface of this chapter, that non-gradient-based optimizers can require a vast amount of function evaluations to converge. However, its implementation is fast and straightforward, making it an excellent first-to-try method to get initial tests on the registration procedures as well as to get baseline results [132]. The BN optimizer requires $(2 \cdot N + 1)$ cost function evaluations per step, defining N as the number of parameters (or dimensions) to optimize. In this case, there are six parameters to optimize (three rotations and three translations), meaning 13 function evaluations per optimization step. Using $\vec{\boldsymbol{p}}$ from (*47*), where each component of the vector represents one of the parameters to optimize, the 13 cost function evaluations can be expressed as to 12 function evaluations and the cost function evaluated at the initial pose $\vec{\boldsymbol{p}}_0 \epsilon \mathbb{R}^6$. The pose at each iteration is interpreted as the initial pose plus an offset vector, and the cost function is evaluated for this sum. This offset vector is added only at one of the parameters per evaluation, and one time is added with a positive sign and other with a negative sign. Mathematically, this offset vector is described as $\vec{\boldsymbol{\Delta}} \epsilon \mathbb{R}^6$, which is shown in the equation (*76*), and the definition of the i^{th} pose to evaluate $\vec{\boldsymbol{p}}_i \epsilon \mathbb{R}^6$ can be found in the equation (*77*).

$$\vec{\boldsymbol{\Delta}}(i) = \begin{cases} [(-1)^i \Delta \quad 0 \quad 0 \quad 0 \quad 0 \quad 0]^T & \boldsymbol{if}\ i = 1,2 \\ [0 \quad (-1)^i \Delta \quad 0 \quad 0 \quad 0 \quad 0]^T & \boldsymbol{if}\ i = 3,4 \\ [0 \quad 0 \quad (-1)^i \Delta \quad 0 \quad 0 \quad 0]^T & \boldsymbol{if}\ i = 5,6 \\ [0 \quad 0 \quad 0 \quad (-1)^i \Delta \quad 0 \quad 0]^T & \boldsymbol{if}\ i = 7,8 \\ [0 \quad 0 \quad 0 \quad 0 \quad (-1)^i \Delta \quad 0]^T & \boldsymbol{if}\ i = 9,10 \\ [0 \quad 0 \quad 0 \quad 0 \quad 0 \quad (-1)^i \Delta]^T & \boldsymbol{if}\ i = 11,12 \end{cases} \tag{76}$$

$$\vec{\boldsymbol{p}}_i = \vec{\boldsymbol{p}}_0 + \vec{\boldsymbol{\Delta}}(i) \tag{77}$$

where:
$i \in \{1,\dots,12\}$
$\vec{\boldsymbol{p}}_0$ is the initial pose
$\vec{\boldsymbol{p}}_i$ is the i^{th} pose to be evaluated.
Δ is the offset added to the initial pose.
$\vec{\boldsymbol{\Delta}}(i)$ is the offset vector describing which parameter is changing.

Now assuming a cost function $f_c(\vec{\boldsymbol{p}}^{(k)})$ with $\vec{\boldsymbol{p}}^{(k)} \epsilon \mathbb{R}^6$, the 13 poses will be evaluated for the iteration k. The cost function value evaluated at $\vec{\boldsymbol{p}}_0$, i.e., $f_c(\vec{\boldsymbol{p}}_0^{(k)})$, is considered as the best value at the beginning of the step k. A new step is executed only, i.e., $k+1$, if at least one of the cost function evaluations $f_c(\vec{\boldsymbol{p}}_1^{(k)})$ to $f_c(\vec{\boldsymbol{p}}_{12}^{(k)})$ is smaller than $f_c(\vec{\boldsymbol{p}}_0^{(k)})$. The pose with the smallest cost function value is promoted as the $\vec{\boldsymbol{p}}_0^{(k+1)}$, see equation (*78*). A new set of evaluations is created and compared once again. When there is no better value in a step than $f_c(\vec{\boldsymbol{p}}_0^{(k)})$, the optimization stops, and the optimized result is $\vec{\boldsymbol{p}}_0^{(k)}$.

$$\begin{aligned}
&While(true):\\
&\quad if\ min\left(\left\{f_c\left(\vec{\boldsymbol{p}}_1^{(k)}\right), \dots, f_c\left(\vec{\boldsymbol{p}}_{12}^{(k)}\right)\right\}\right) \ < \ f_c\left(\vec{\boldsymbol{p}}_0^{(k)}\right):\\
&\quad then: \quad i = \underset{1\le i\le 12}{argmin} \ \ f_c\left(\vec{\boldsymbol{p}}_i^{(k)}\right)\\
&\qquad\qquad\quad \vec{\boldsymbol{p}}_0^{(k+1)} = \vec{\boldsymbol{p}}_i^{(k)}\\
&\quad else: \quad \underset{\vec{\boldsymbol{p}}\epsilon\mathbb{R}^6}{argmin} \ f_c(\vec{\boldsymbol{p}}) \ = \vec{\boldsymbol{p}}_0^{(k)}\\
&\qquad\qquad\quad break;
\end{aligned} \tag{78}$$

BN is similar in implementation to the Downhill-Simplex method [133], with the difference that Downhill-Simplex uses $(N+1)$ evaluations per step.

Gradient Descent (GRDE)

As its name indicates, GRDE uses the gradient of the cost function to find its minimum. From an initial pose, $\vec{\boldsymbol{p}}_{\mathbf{0}}$, the gradient, $\vec{\nabla}_{\boldsymbol{p}} f_c(\vec{\boldsymbol{p}}_{\mathbf{0}})$, is calculated using the equation (*74*). The gradient indicates the slope of the function at that point. Assuming that the function is convex, or at least in the neighborhood of the minimum, it is logical to move into the next step proportionally to the gradient [132]. In gradient descent, this movement is regulated by a learning rate, λ, which is set experimentally based on the problem [134]. The GRDE updating rule is described by equation (*79*).

$$\begin{aligned}
&While(true):\\
&\quad if\ \ f_c\left(\vec{\boldsymbol{p}}^{(k)} - \lambda \cdot \vec{\nabla}_{\boldsymbol{p}} f_c\big(\vec{\boldsymbol{p}}^{(k)}\big)\right) \ < \ f_c\big(\vec{\boldsymbol{p}}^{(k)}\big):\\
&\quad then: \quad \vec{\boldsymbol{p}}^{(k+1)} = \vec{\boldsymbol{p}}^{(k)} - \lambda \cdot \vec{\nabla}_{\boldsymbol{p}} f_c\big(\vec{\boldsymbol{p}}^{(k)}\big)\\
&\quad else: \quad \underset{\vec{\boldsymbol{p}}\epsilon\mathbb{R}^6}{argmin} \ \ f_c(\vec{\boldsymbol{p}}) \ = \vec{\boldsymbol{p}}^{(k)}\\
&\qquad\qquad\quad break;
\end{aligned} \tag{79}$$

After finding the next pose, $\vec{\boldsymbol{p}}^{(k+1)}$, the cost function is evaluated, and the value is compared with $f_c\big(\vec{\boldsymbol{p}}^{(k)}\big)$. If the cost function value of the new pose, $f_c\big(\vec{\boldsymbol{p}}^{(k+1)}\big)$is smaller than the value of the initial pose, $f_c\big(\vec{\boldsymbol{p}}^{(k)}\big)$, a new step is carried out. Otherwise, the iterative process stops, and the initial pose of the step is considered as the minimum of the function, i.e., $\vec{\boldsymbol{p}}^{(k)}$.

AdaGrad (AG)

AG is an updated version of gradient descent, which adapts the learning rate, λ, of the optimizer for each step and each parameter. AG achieves the learning rate adaption by accumulating the sum of squares of the past gradients. The learning rate is then divided by the squared root of the past gradients. The accumulation is done individually per each parameter and represented in the vector $\vec{\boldsymbol{g}}_t\epsilon\mathbb{R}^m$ where m

refers to the number of parameters, and each component contains the gradient accumulation of the m^{th} parameter [135]. In this case $m = 6$ and $\vec{\boldsymbol{g}}_t \epsilon \mathbb{R}^6$. The AG updating equation is presented in equation (*80*).

$$
\begin{aligned}
&While(true): \\
&\vec{\boldsymbol{g}}_t^{(k)} = \sum_{i=1}^{k} \vec{\nabla}_{\boldsymbol{p}} f_c\left(\vec{\boldsymbol{p}}^{(i)}\right)^2 \\
&if\ \ f_c\left(\vec{\boldsymbol{p}}^{(k)} - \frac{\lambda \cdot \vec{\mathbf{1}}}{\sqrt{\vec{\boldsymbol{g}}_t^{(k)} + \varrho \cdot \vec{\mathbf{1}}}} \cdot \vec{\nabla}_{\boldsymbol{p}} f_c\left(\vec{\boldsymbol{p}}^{(k)}\right)\right) < f_c\left(\vec{\boldsymbol{p}}^{(k)}\right): \\
&then: \quad \vec{\boldsymbol{p}}^{(k+1)} = \vec{\boldsymbol{p}}^{(k)} - \frac{\lambda \cdot \vec{\mathbf{1}}}{\sqrt{\vec{\boldsymbol{g}}_t^{(k)} + \varrho \cdot \vec{\mathbf{1}}}} \cdot \vec{\nabla}_{\boldsymbol{p}} f_c\left(\vec{\boldsymbol{p}}^{(k)}\right) \\
&else: \quad \underset{\vec{\boldsymbol{p}} \epsilon \mathbb{R}^6}{argmin}\ f_c(\vec{\boldsymbol{p}}) = \vec{\boldsymbol{p}}^{(k)} \\
&\qquad\quad break;
\end{aligned}
\tag{80}
$$

The vector $\vec{\mathbf{1}} \epsilon \mathbb{R}^6$ is a column vector of all ones. The term ϱ is added to avoid division by zero and is usually set to a small constant, e.g., 1×10^{-8}. It is worth to notice that $\vec{\boldsymbol{g}}_{\mathrm{t}}$ keeps growing after each iteration, which forces the learning rate to decrease. This behavior makes the update of the $\vec{\boldsymbol{p}}^{(k+1)}$ each time smaller up to a point, where the updates have no effect on the estimate. This problem is addressed in the next optimizer.

AdaDelta (AD)

AD is a parallel implementation of AG that intents to reduce its monotonic decreasing learning rate. It is achieved by accumulating the squared of the past gradients as a decaying average, also known as running average, $\vec{\boldsymbol{r}}_a\left(\vec{\nabla}_{\boldsymbol{p}} f_c\left(\vec{\boldsymbol{p}}^{(k)}\right)^2\right) \epsilon \mathbb{R}^6$. See equation (*81*). The decay is controlled by term γ usually set to 0.9.

$$
\vec{\boldsymbol{r}}_a\left(\vec{\nabla}_{\boldsymbol{p}} f_c\left(\vec{\boldsymbol{p}}^{(k)}\right)^2\right) = \gamma \cdot \vec{\boldsymbol{r}}_a\left(\vec{\nabla}_{\boldsymbol{p}} f_c\left(\vec{\boldsymbol{p}}^{(k-1)}\right)^2\right) + (1-\gamma) \cdot \vec{\nabla}_{\boldsymbol{p}} f_c\left(\vec{\boldsymbol{p}}^{(k)}\right)^2 \tag{81}
$$

The running average of the element-wise square parameters, $\vec{\boldsymbol{r}}_a\left({\vec{\boldsymbol{p}}^{(k)}}^2\right) \epsilon \mathbb{R}^6$, is additionally introduced in equation (*82*)

$$
\vec{\boldsymbol{r}}_a\left({\vec{\boldsymbol{p}}^{(k)}}^2\right) = \gamma \cdot \vec{\boldsymbol{r}}_a\left({\vec{\boldsymbol{p}}^{(k-1)}}^2\right) + (1-\gamma) \cdot \left({\vec{\boldsymbol{p}}^{(k)}}^2\right) \tag{82}
$$

The vector $\vec{\boldsymbol{g}}_t$ in equation (*80*) is changed by the running average of gradients from equation (*81*). The learning rate, λ, is changed by the running average of parameters from equation (*82*) [136]. The update rule can be seen in equation (*83*)

$$
\begin{aligned}
&While(true):\\
&if\ \ f_c\left(\vec{\boldsymbol{p}}^{(k)} - \frac{\vec{\boldsymbol{r}}_a\left(\vec{\boldsymbol{p}}^{(k-1)^2}\right)}{\sqrt{\vec{\boldsymbol{r}}_a\left(\vec{\nabla}_{\boldsymbol{p}} f_c\left(\vec{\boldsymbol{p}}^{(k)^2}\right)^2\right) + \varrho}} \cdot \vec{\nabla}_{\boldsymbol{p}} f_c\left(\vec{\boldsymbol{p}}^{(k)}\right)\right) < f_c\left(\vec{\boldsymbol{p}}^{(k)}\right):\\
&then: \quad \vec{\boldsymbol{p}}^{(k+1)} = \vec{\boldsymbol{p}}^{(k)} - \frac{\vec{\boldsymbol{r}}_a\left(\vec{\boldsymbol{p}}^{(k-1)^2}\right)}{\sqrt{\vec{\boldsymbol{r}}_a\left(\vec{\nabla}_{\boldsymbol{p}} f_c\left(\vec{\boldsymbol{p}}^{(k)^2}\right)^2\right) + \varrho}} \cdot \vec{\nabla}_{\boldsymbol{p}} f_c\left(\vec{\boldsymbol{p}}^{(k)}\right)\\
&else: \quad \underset{\vec{\boldsymbol{p}} \epsilon \mathbb{R}^6}{argmin}\ f_c(\vec{\boldsymbol{p}}) = \vec{\boldsymbol{p}}^{(k)}\\
&\qquad\quad break;
\end{aligned}
\tag{83}
$$

An additional advantage of AD over AG is that it does not require a learning rate.

Adaptive Moment Estimation (Adam)

Adam is another optimizer that uses gradients and computes adaptive learning rates for each parameter [137]. The learning rate is updated based on the concept of momentum. It gives the gradients a physical concept of a sphere with heavy mass. The optimization is seen as the sphere going down a hill with friction. Consequently, in case of facing a mound when going down, the sphere can climb it and keep going down due to its inertia. This concept helps to overcome local minima and gives preferences to flat minima in error surfaces [138]. Let the decaying average of gradients and squared gradients be defined as $\vec{\boldsymbol{m}}^{(k)} \epsilon \mathbb{R}^6$ and $\vec{\boldsymbol{v}}^{(k)} \epsilon \mathbb{R}^6$. Equation (*84*) describes $\vec{\boldsymbol{m}}^{(k)}$ and $\vec{\boldsymbol{v}}^{(k)}$, which are estimations of the first and second moments of the gradients, respectively.

$$
\begin{aligned}
\vec{\boldsymbol{m}}^{(k)} &= \beta_1 \cdot \vec{\boldsymbol{m}}^{(k-1)} + (1-\beta_1) \cdot \vec{\nabla}_{\boldsymbol{p}} f_c\left(\vec{\boldsymbol{p}}^{(k)}\right)\\
\vec{\boldsymbol{v}}^{(k)} &= \beta_2 \cdot \vec{\boldsymbol{v}}^{(k-1)} + (1-\beta_2) \cdot \vec{\nabla}_{\boldsymbol{p}} f_c\left(\vec{\boldsymbol{p}}^{(k)}\right)^2
\end{aligned}
\tag{84}
$$

With these estimations, the update rule can be computed as equation (*85*) shows. The change in the current step is proportional to the first moment, $\vec{\boldsymbol{m}}^{(k)}$, divided by the squared root of the second moment, $\vec{\boldsymbol{v}}^{(k)}$, and regulated by the learning rate λ.

$$
\begin{aligned}
&While(true):\\
&if\ \ f_c\left(\vec{\boldsymbol{p}}^{(k)} - \frac{\lambda \cdot \vec{\mathbf{1}}}{\sqrt{\vec{\boldsymbol{v}}^{(k)} + \varrho \cdot \vec{\mathbf{1}}}} \cdot \vec{\boldsymbol{m}}^{(k)}\right) < f_c\left(\vec{\boldsymbol{p}}^{(k)}\right)\\
&then: \quad \vec{\boldsymbol{p}}^{(k+1)} = \vec{\boldsymbol{p}}^{(k)} - \frac{\lambda \cdot \vec{\mathbf{1}}}{\sqrt{\vec{\boldsymbol{v}}^{(k)} + \varrho \cdot \vec{\mathbf{1}}}} \cdot \vec{\boldsymbol{m}}^{(k)}\\
&else: \quad \underset{\vec{\boldsymbol{p}} \epsilon \mathbb{R}^6}{argmin}\ f_c(\vec{\boldsymbol{p}}) = \vec{\boldsymbol{p}}^{(k)}\\
&\qquad\quad break;
\end{aligned}
\tag{85}
$$

The terms β_1 and β_2 are suggested by the literature to be set as 0.9 and 0.999, respectively [137].

4.7.4. 2D/3D Intensity-Based Registration Procedure

The 2D to 3D registration procedure is a logical interconnection of different subtasks aiming to find the transformation between a 3D volume and a 2D image. A CT-scan is projected as a 2D image, as explained in section 4.7.1, and the DRR image is compared with an X-ray image using a merit function, as seen in section 4.7.2. In the end, an optimizer, as shown in section 4.7.3, is used to find the CT-scan pose with respect to the {DRR}, equation (*47*), that matches the projection of the 3D image with the 2D image. As said before, the 2D/3D registration is an iterative process that keeps creating a DRR image based on the new pose calculated by the optimization algorithm and compares it with the X-ray. The process is represented in the block diagram of Figure 4-27. Notice the loop *DRR - merit function - optimization* that only breaks when the registration is over.

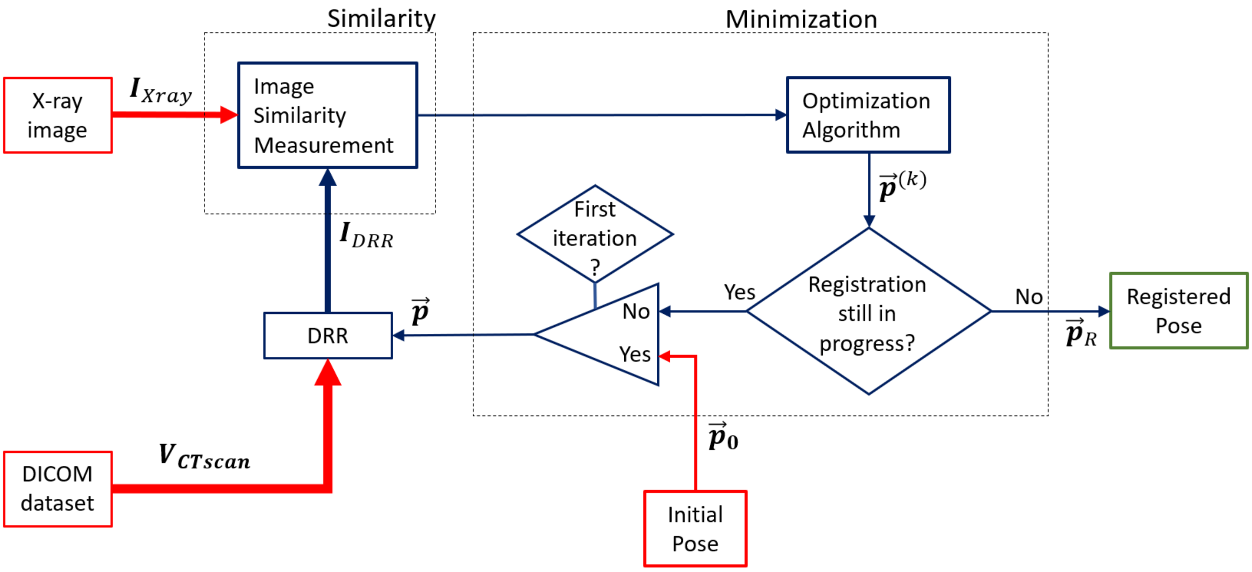

Figure 4-27. Block diagram of the 2D to 3D intensity-based registration

The registration process is represented mathematically in equation (*86*) by direct application of (*75*). The names used in the function also match the names in Figure 4-27,

$$\vec{p}_R = \underset{\vec{p}\in\mathbb{R}^6}{argmin}\left(f_c\left(I_{Xray}, V_{CTscan}, \vec{p}\right)\right) \qquad (86)$$

using $\vec{p}_0$ as the initial pose for the registration process, and obtaining $\vec{p}_R$ as the registered pose. The implementation of the registration process is developed in sections 5.1 and 5.2.

4.8. Deep-Learning Applied to the 2D/3D Registration

During the last years, artificial intelligence (AI), more specifically deep-learning, has become a powerful ally of the medical field. Some applications such as diagnosis, medical image enhancement, and disease prediction and control are examples of notable deep-learning contributions. As time has passed, computational power has been increasing exponentially. The amount of publicly accessible data expands every year, e.g., Microsoft Common Objects in Context (COCO) dataset, Open Images dataset, PASCAL VOC (Visual Object Classes) dataset, ImageNet dataset, among others. The combination of those facts has raised the interest in the development of data-driven approaches that make use of significant amounts of data and high-performance computational hardware. An approach of AI is machine learning that uses statistical techniques to give learning capabilities to computers. One of the most typical structures within machine learning is the artificial neural network, which uses an underlying structure called a neuron. The neurons are interconnected with each other, forming a mesh, as the biological neuronal circuitry found in the brain, to learn complex patterns. However, machine learning approaches require some expertise to preprocess the data, label datasets, and handcraft the relevant features before they can be used. It is worth saying that the results are highly dependent on the handcrafted features and annotations quality. Here is when the deep-learning approaches play an interesting role. Deep-Learning models are end-to-end learning algorithms, which autonomously learn to extract relevant features. Those models require a massive amount of data to be adequately trained, but it finally depends on the network size and complexity.

One of the models that boosted the deep-learning approaches in the image processing field was the development of the *convolutional neural network* (CNN). Its fundamental component is a convolutional kernel, which executes a convolution operation over the input image. By design in CNNs, the size of the convolution kernels is fixed, but the kernel values are learned during training. A representation of the convolution operation over an image is depicted in Figure 4-28.

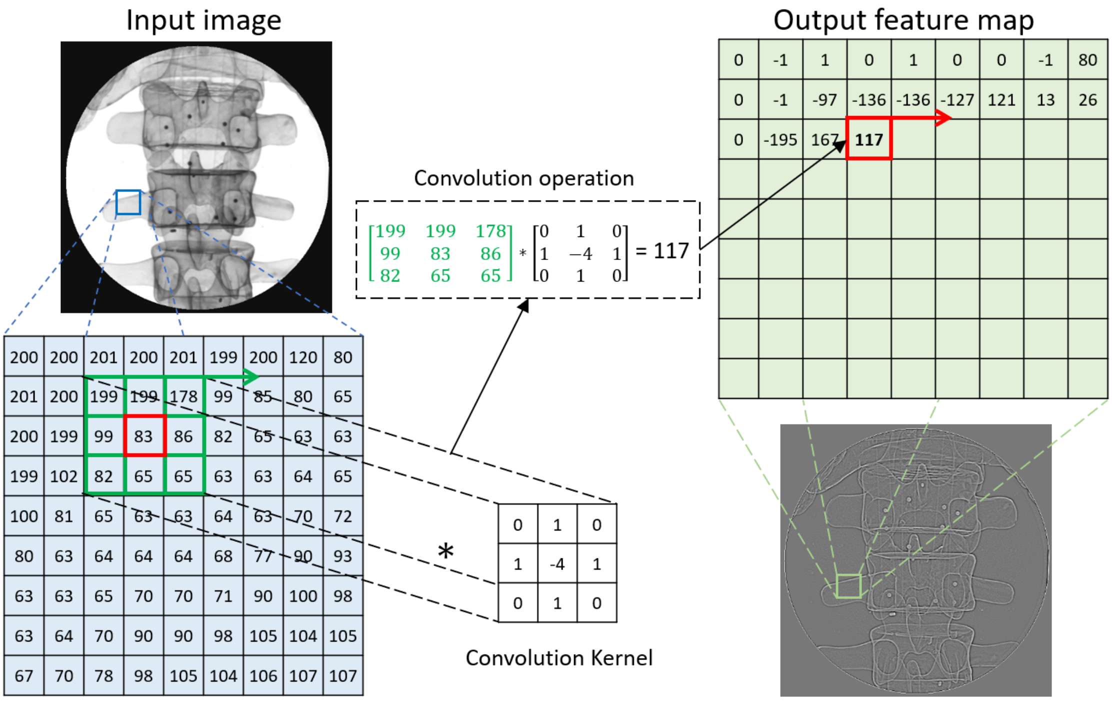

Figure 4-28. Convolution operation on an image

Two parameters drastically change the convolution operation output, namely stride and padding. The *stride* controls the sliding behavior of the kernel over the input image. It means the kernel can be set to move a specific number of elements over the image in each step. It is worth noting that a stride different than one will reduce the output layer size. Additionally, the kernel movement is delimited by the input image edges. A simple approach consists of restricting the kernel movement, such as the kernel edges do not overpass the input edges. In this case, the output image is reduced in dimension as the kernel center does not move within the entire input. To keep the input size, an artificial border, usually filled with zeros, is added to the input, such as the central kernel element can move around all the input elements. Enabling the kernel movement by extending the input edge is called *padding*. It is usually set such as it preserves the size of the output feature map and uses the information on the image corners. Changes in size are usually executed by a particular layer called *pooling*, added after convolution layers. A pooling kernel of size $n \times n$ reduces the image size by n times. A typical size for a pooling kernel is 2×2.

The convolution kernels are the CNN portion extracting the relevant features for the application. It is common to use a fully connected artificial neural network (FCNN) after the cascade of convolutions and pooling layers. The FCNN is in charge of learning patterns from those selected features and infers results. The output nodes of FCNN and convolutions layers are connected to activation functions, adding non-linearities. These non-linear functions enable artificial neural networks to compute non-trivial problems. The selection of an activation function for a particular application is based on trends in the literature. The hyperbolic tangent (Tanh) was years ago a widely used activation function option, but nowadays, the rectified linear unit (ReLU) is one of the most popular choices for an activation function as it shows better results than the former [139].

Two concepts must be addressed in deep-learning, overfitting and underfitting. They are used to describe the representation of the problem by a deep-learning model. The main goal of deep-learning approaches is to build learning models that can generalize unseen scenarios. If the model adjusts perfectly to the training data instead of learning patterns, the model is *overfitted*. An overfitted model fails when it is subjected to unseen data. Conversely, *underfitting* describes models that perform poorly on training data. A proper choice of model capacity can control its ability to fit the task and avoid overfitting and underfitting problems [140].

One of the most time-consuming tasks in solving a problem with a deep-learning approach is finding the model architecture. This is commonly addressed using one well-known model architecture, e.g., ResNet, VGGNet, Inception, which has shown outstanding results in a similar application. Another common practice is fine-tuning a pre-trained model to learn a new task instead of training them from scratch. Unlike training from scratch, fine-tuning does not require high computational power or vast data for model training. This approach is used to develop the initial pose generator. See Section 7.2.

CNNs bring the advantage of achieving a self-feature extraction behavior, which requires a cascade of several layers. For instance, DenseNet-BC uses several hundreds of sequential convolutions, classifying an object within 100 categories with an accuracy of 96.54% [141]. The entire layout of a CNN self-extracts numerous features by connecting the output of one convolution kernel with a new one. It starts extracting edges, which are combined to distinguish objects, and deeper in the CNN, these learning objects are used to detect big-end features such as faces or people. As the convolution is an operation that moves over the entire image, the prediction contains spatial robustness, meaning the system can detect a specific pattern regardless of the location within the image [142].

In medicine, CNNs have been used in the diagnosis of cancer, tuberculosis, cataract, and tumors, among others [143] [144] [145] [146]. One application that has fostered the use of deep-learning is the registration of medical modalities. 2D to 3D registration focusing on Total Knee Arthroplasty kinematics, Virtual implant planning system, and X-ray echo fusion have introduced deep-learning into the registration process [147] [148]. In these works, intensity-based registration is still used, employing DRR and X-ray images. The deep-learning adaptation is made by including a CNN as a replacement of the merit function and the optimizer. In other words, the system is still an iterative process, and the X-ray image and DRR are the CNN inputs. The CNN infers the following pose for the DRR. Similarly, another study focused on the registration of brain images using a CNN to replace the merit function and the optimizer [149].

Another application of 2D to 3D registration is tracking lung deformation during the exhaling and inhaling phase for use in radiotherapy. A deformation set matrix expresses the change in the lungs between images. A CNN architecture is used in that study to compute the coefficients of the deformation matrix [150]. In another application for lungs, where pre-operative CT data is registered with X-ray images, a CNN is used for tracking and triangulation using a point-of-interest network. This deep-learning approach manages to find correlated points in the X-ray images and the CT-scan. After those points are calculated, the registration is calculated analytically [151].

It is worth noting that up to the moment that this work was written, no research was found in 2D to 3D registration focusing on spine applications. Some implementations of deep-learning to improve the 2D to 3D registration are developed in chapter 7.

5. Evaluation and Optimization of the Selected 2D/3D Image Registration Approach

5.1. Reference Frames Involved in the 2D to 3D Registration

Systems with multiple reference frames require to bring elements, e.g., vectors and points, to a common coordinate system. Once the elements are in the same coordinate system, operations among them can be carried out. Then it is necessary to know the transformations among references to change elements from one frame to another. In the case of the 2D/3D registration, there are several reference frames involved. Some of them are auxiliary frames created to facilitate calculations, while others have a physical meaning. Commonly, rigid bodies (RB) define a physical frame, which a navigation system recognizes. The navigation system is a stereo vision sensor, containing an accuracy component that must be understood to determine the maximum theoretical accuracy of the registration.

This section presents an overview of rigid bodies, coordinate systems, and navigation system properties. As stated in chapter 3, the number of foreign objects in the operating room (OR) should be kept to a minimum. Some considerations about the number of used RBs are given, and a diagram of the used reference frames in the developed 2D/3D registration is shown. A complete picture of the involved coordinate systems gives a clear idea about the number of transformations that must be calculated before reaching a useful result. In other words, the equation of the 2D/3D registration is calculated, but there is no discussion about the meaning of the equation terms. The method to obtain each of those transformations is explained individually in the coming sections. It means the objectives of each of the following chapters are contextualized within a single term of the 2D/3D registration equation.

5.1.1. Considerations of the Navigation System Measurements

During the development of this work, an NDI Vega optical navigation system is used. The specifications given by the manufacturer are analyzed to find the accuracy of the reported poses. Since two cameras compose the navigation system, the measures are limited to a range in the space. Both cameras must see the element of interest, and it must be close enough to create pixel disparity between images [152]. This range, where the measurements are valid, is known as the measurement volume. The entire volume is shown in Figure 5-1. It is split into an accurate pyramid volume in green and a less precise extended volume in blue.

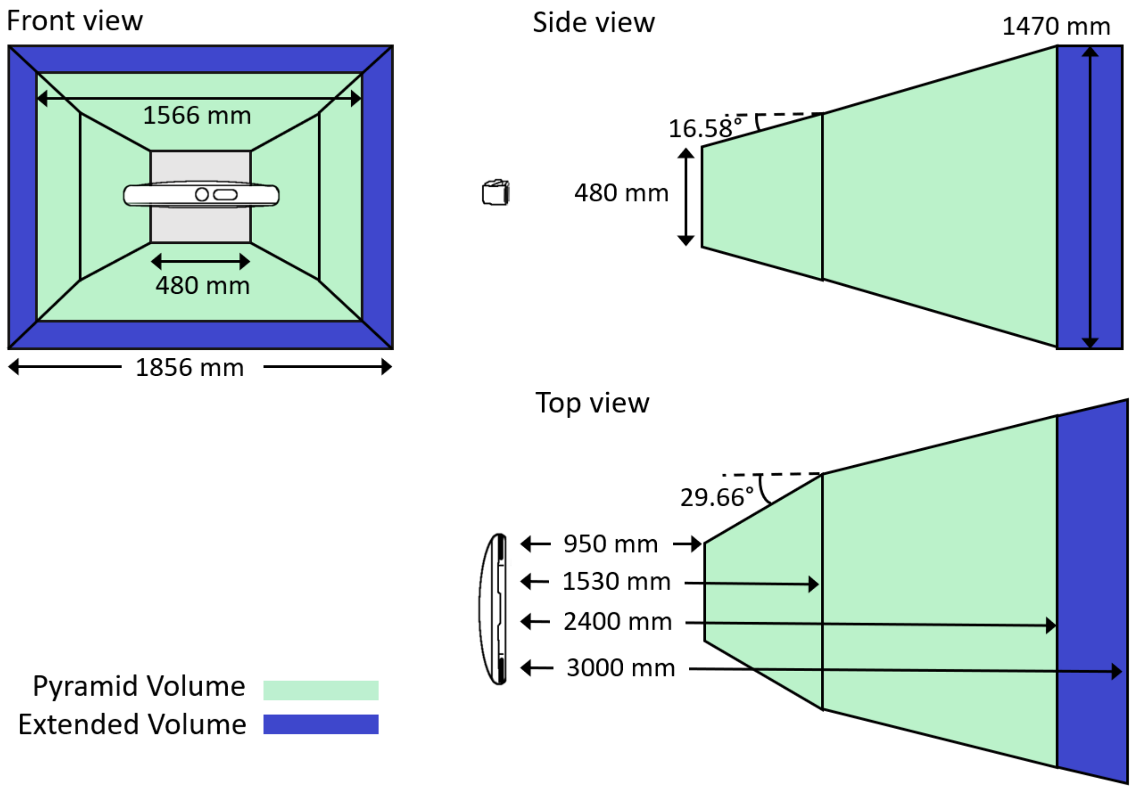

Figure 5-1. NDI Vega measurement volume

The manufacturer guarantees that the RMS values of accuracy and repeatability within the pyramid volume are less than or equal to 0.25mm and 0.10mm, respectively [153]. The manufacturer suggests using the navigation system only for relative measurements, i.e., the measurement of an RB of interest is taken with respect to another RB. Contrary to absolute measurements, when the measurement of an RB of interest is considered directly with respect to the camera reference frame. When the relative measurements are used, the repeatability error is the acting source of error; consequently, a more accurate measurement is guaranteed.

Only relative measurements are used in this work, so the correct error to consider is the repeatability error. It can be said that the navigation system accuracy is 0.1mm when the measurements are under the pyramid volume.

5.1.2. Reference frames involved in the 2D/3D Registration

The RBs that are tracked by the navigation system have pre-defined reference frames, which are loaded into the camera as definition files (one definition file per RB). The reported pose of the RB is the transformation from the navigation system reference frame to the RB reference frame. The navigation coordinate frame, hereafter referred to as the optical tracking system (OTS), is depicted in Figure 5-2. In short, the navigation system reports the transformation ${}^{OTS}\boldsymbol{T}_{RB}$.

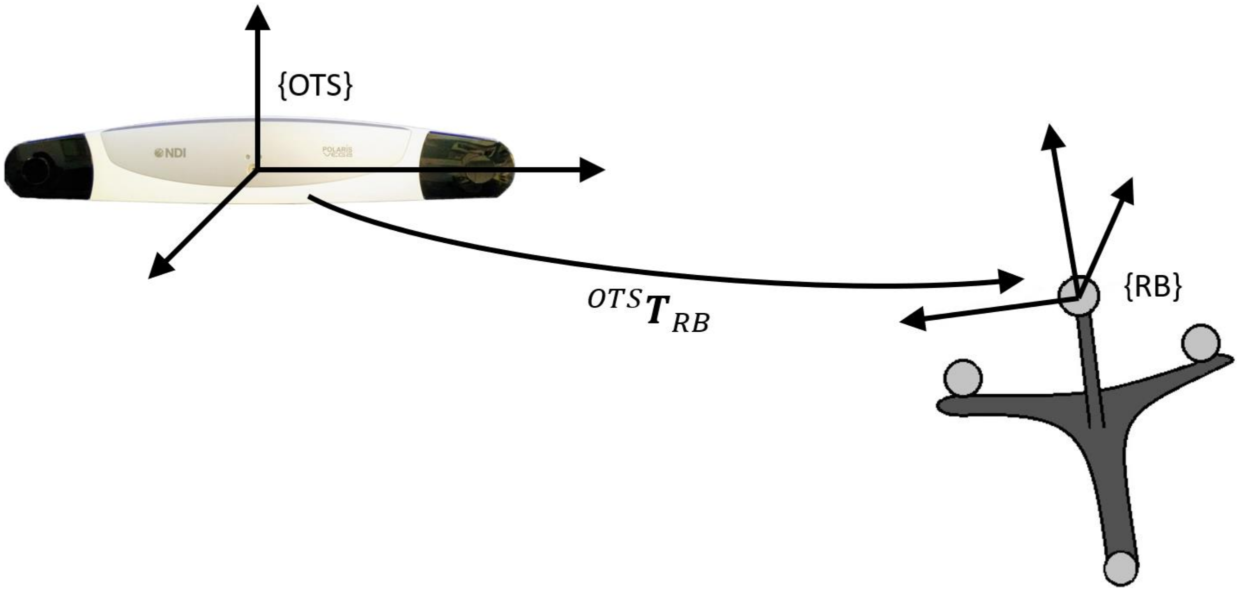

Figure 5-2. NDI Vega reference frame and reported transformation with RB

Taking advantage of the RBs feedback from the navigation system, it seems intuitive to use as many as possible to facilitate the registration process. However, it is essential to limit the use of foreign elements inside the OR. It means that the number of RBs must be limited to a minimum. As part of the 2D/3D registration, only two RBs will be used. One RB is used as the reference frame for the C-arm, CRB, which stands for C-arm Rigid Body. A second RB is attached to the patient's anatomy, referred to as ARB, which stands for Aim Rigid Body. The reference frames created by those RBs are called {CRB} and {ARB}, respectively. The {CRB} frame, despite its similar name, should not be confused with {C-ARM}. {CRB} is a frame created from a physical device while {C-ARM} is an auxiliary reference frame used to describe the X-ray source of the C-arm. The frames involved in the 2D/3D registration are depicted in Figure 5-3.

A third RB is attached to the robot end-effector, helping to track the robot tool, a drill bit for pedicle screw insertion. This third RB is known as the Tool Rigid Body, TRB, which plays no role during the registration process, but it is necessary during the robot-assisted surgery. In addition to those reference frames, the camera frame {OTS} is also considered.

In the computer, a planning software displays the 3D modality taken by a CT-scanner. This planning software aids the physician in evaluating the patient's condition and plans the screw insertion. In this work, the reference frames that have meaning only in a digital environment are known as virtual reference frames. 3D modalities have the typical medical format for images, DICOM format, which has its virtual reference frame {DICOM}. Every implant planned on the planning software has coordinates on the {DICOM} frame.

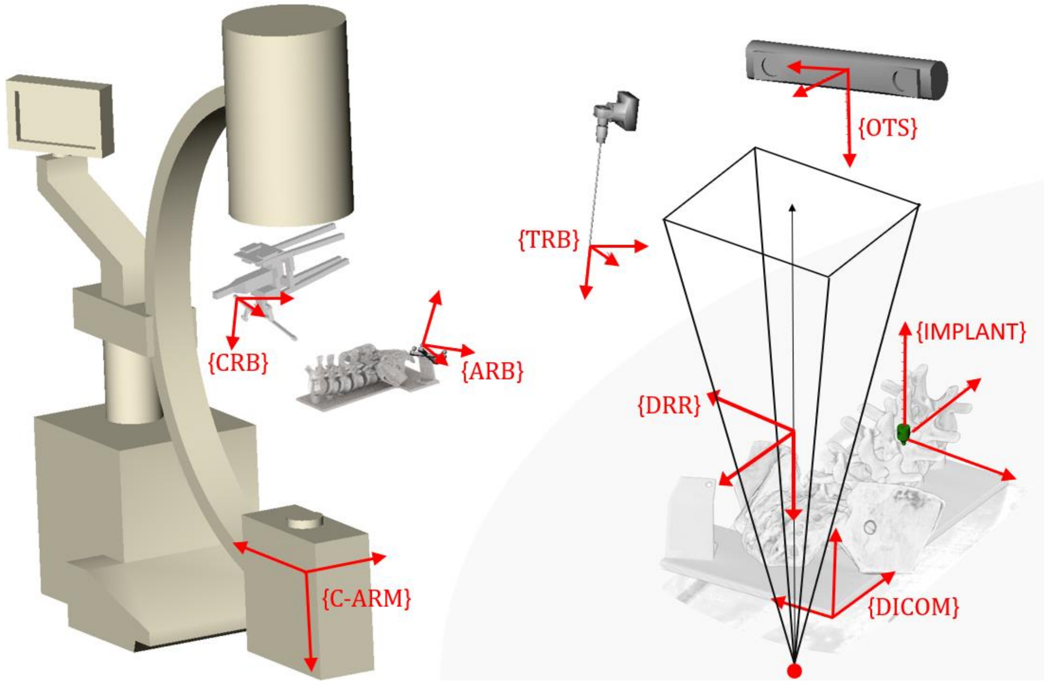

Figure 5-3. Frames involved in the 2D/3D registration

To execute the surgical procedure, the robot control requires the target coordinates to be associated with its external RB, {TRB}. The patient's data is transformed to the patient frame, {ARB}, as ARB is rigidly attached to the patient. Then the robot can execute a pose of the type ${}^{TRB}\boldsymbol{T}_{ARB}$ as the target coordinate. To connect the planned (pre-operative) data, which is done in the {DICOM} frame, the very fundamental goal of the registration process would be to find the transformation from {DICOM} to {ARB} frame, ${}^{ARB}\boldsymbol{T}_{DICOM}$. This transformation closes the path to send coordinates in {DICOM} to the robot end-effector {TRB}.

To find the transformation ${}^{ARB}\boldsymbol{T}_{DICOM}$, it is necessary to use auxiliary frames. A mere mathematical representation of the reference frames involved in the 2D/3D registration can be seen in Figure 5-4. In the diagram, {TRB} is omitted as it is not used for the registration procedure, and an additional reference frame {DRR} is introduced. In section 4.7.1, it is explained the digitally reconstructed radiograph (DRR) algorithm. The DRR, which makes operations on the 3D modality, has a reference frame; therefore, this module is defined by the virtual reference frame {DRR}.

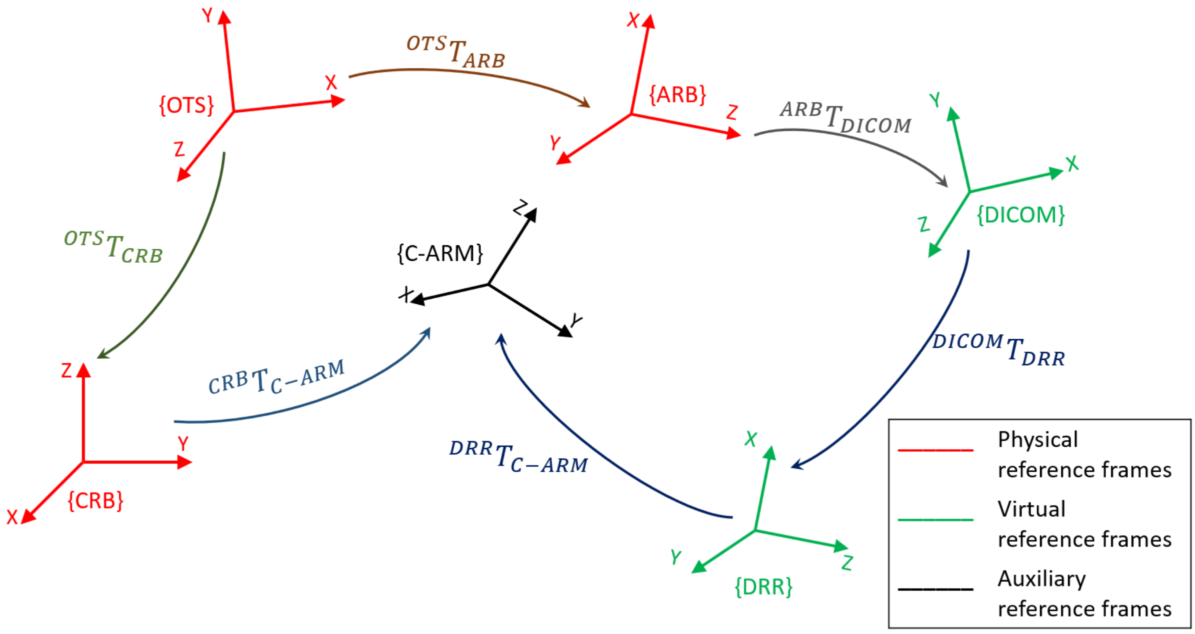

Figure 5-4. Mathematical representation of frames involved in the 2D/3D registration

5.1.3. Equation of the 2D/3D registration

As the primary purpose of the registration is finding the transformation $^{ARB}\boldsymbol{T}_{DICOM}$, it is possible to use Figure 5-4 to define the chain of transformations. Starting on {ARB}, it is possible to move back to {OTS} and from there to {CRB} using the navigation system feedback. The next step is moving from {CRB} to the auxiliary {C-ARM} frame. In the step from {C-ARM} to {DRR}, the transition from physical reference frames to virtual frames is carried out. The final transformation is done within the virtual frames, from {DRR} to {DICOM}. Equation *(87)* shows the set of transformations described above.

$$^{ARB}\boldsymbol{T}_{DICOM} = {}^{ARB}\boldsymbol{T}_{OTS} \cdot {}^{OTS}\boldsymbol{T}_{CRB} \cdot {}^{CRB}\boldsymbol{T}_{C-ARM} \cdot {}^{C-ARM}\boldsymbol{T}_{DRR} \cdot {}^{DRR}\boldsymbol{T}_{DICOM} \tag{87}$$

The first two terms of the equation *(87)*, i.e., $^{ARB}\boldsymbol{T}_{OTS} \cdot {}^{OTS}\boldsymbol{T}_{CRB}$, are known transformations reported by the navigation system, so there is no need for further development on these two terms. The third element of the equation, namely, $^{CRB}\boldsymbol{T}_{C-ARM}$, represents the transformation from {CRB} to the C-arm X-ray source. This transformation is found by the parametrization of the C-arm, as explained in section 4.5.4. The specifics of the implementation are shown in detail in section 5.4. The fourth term of the equation *(87)*, $^{C-ARM}\boldsymbol{T}_{DRR}$, explains the connection between the C-arm and the DRR module, which is the transition from real to virtual reference frames. This transition is clarified in section 5.2. The final term, $^{DRR}\boldsymbol{T}_{DICOM}$, results from the DRR module when the optimization process is complete. The optimization for the 2D/3D registration is explained in section 4.7.3, and the actual implementation is discussed in section 5.5.

5.1.4. Discussion and Conclusions

Analyzing the information given by the navigation system manufacturer, it is found that the NDI Vega system can give RMS measurement errors as good as 0.1mm when using the pyramid volume.

The maximum number of RBs being used for the 2D/3D registration is limited to two, which is the minimum of RBs necessary to compute the 2D/3D registration and to fulfill the requirement to minimize the number of RBs in the operating room. A third RB on the robotic system, though, is required to execute the surgical procedure.

The reference frames used for the 2D/3D registration are established and depicted in Figure 5-4. From there, it is possible to determine the transformation to bring the pre-operative data into the operation theater, i.e., ${}^{ARB}\boldsymbol{T}_{DICOM}$.

The equation of the 2D/3D registration, expressed by equation (*87*), describes the transformations required to find ${}^{ARB}\boldsymbol{T}_{DICOM}$.

5.2. Basic Structure of the 2D/3D Registration Using Intra-Operative X-ray Images and Pre-operative CT-Scan

In this chapter, a more detailed explanation about the interconnection of internal modules is given, and the transformation ${}^{DRR}\boldsymbol{T}_{DICOM}$ is explained in detail. Additionally, some considerations regarding the feedback of the navigation system are explained. From equation *(87)*, it is deduced that the registration results are transformed into a useful frame with the help of the camera feedback. When using one image to execute the registration, the camera feedback is not involved during the optimization process that carries out the registration. When two or more images are used for the registration, the camera feedback plays an important role. The connection between X-rays is achieved by finding the transformations from the C-arm on position A with the C-arm in position B, ${}^{C-ARM_{\mathrm{A}}}\boldsymbol{T}_{C-ARM_{\mathrm{B}}}$. For example, an image with anterior-posterior (AP) projection and another with lateral (LAT) projection can be used for the registration once the registration ${}^{C-ARM_{\mathrm{A}}}\boldsymbol{T}_{C-ARM_{\mathrm{B}}}$ is known. In other words, ${}^{C-ARM_{\mathrm{A}}}\boldsymbol{T}_{C-ARM_{\mathrm{B}}}$ is the transformation that incorporates both images into the registration process.

Furthermore, the idea of the region of interest (ROI) is introduced, which delimits the X-ray image area to be registered. The main idea of the ROI is to focus the registration on important structures so that the ROI can be selected individually in every image used for the registration. The output of the image after the ROI is used as the input for the merit function. In the case of two images, two merit functions act simultaneously, but the optimizer still works in the same manner. The value given to the optimizer is the weighted average of the merit functions. As the ROIs in each image may have different sizes, the ROIs areas are used as the weights.

At the end of this chapter, a new procedure is introduced to automate the computation of the 2D/3D registration accuracy. The registration accuracy is indirectly computed using a landmark in the CT-scan of a testing device. The same landmark is identified in the physical object, and the registration result is used for transforming the landmark from the CT-scan to the testing device. The deviation between the transformed landmark and the measured landmark on the tested device can be calculated. This deviation is used as a metric for the registration quality. The computation of the accuracy can be automated to calculate the error of several registration results. Therefore, the accuracy assessment can be done through hundreds of registrations, which gives a better estimation of the implemented 2D/3D registration process.

5.2.1. Structure of the 2D/3D registration

It has been previously mentioned that the implemented solution is an intensity-based 2D/3D registration. The two main inputs of the registration are a 3D CT-scan in DICOM format and two X-ray images. In theory, the registration can be done with only one X-ray image. It works optimally with two X-rays, but it is also possible to use more than two images. However, some studies have shown no advantage of using more than two X-ray images, as long as the images are taken physically perpendicular [70]. The case with one X-ray image is used as primary base to explain the registration process. The two images case is then introduced and further generalized to make a 2D/3D registration using N X-ray images.

The registration process will compare the X-ray image with several DRR images. The minimization process starts with an initial pose, which is applied to the DRR module. In this work, two novel approaches are taken to generate the initial pose. A manual approach, where the user can create a DRR image from the same perspective as the current CT-scan visualization, is explained in section 5.7. The second approach, an automatically generated initial pose using deep-learning, is developed in detail in section 7.2.

The registration process, introduced in section 4.7.4, consists of an optimizer that finds the minimum of a cost function, which follows the equation (*86*). The cost function has as inputs an X-ray image and a DRR image, and it uses an image similarity measurement, explained in section 4.7.2, to return a numerical output. The updating rules for each optimizer, explained in section 4.7.3, differ from each other, but a common calculation is finding the cost function gradient with respect to the pose.

For calculating the gradient in iteration k , let the starting point be the pose $\vec{\boldsymbol{p}}^{(k)}$. The cost function is then evaluated as it is explained for the best neighbors optimizer, in section 4.7.3, using the equation *(76)* and (*77*). *Δ* in equation (*76*) is set to 0.1mm for $1 \leq i \leq 6$ and set to 0.1° for $7 \leq i \leq 12$. The derivative for any specific cost function parameter is approximated by following the symmetric (central) difference quotient. The literature shows that a better approximation of the derivative is given by the symmetric difference quotient than one-side quotient approximations when using a small *Δ* [154]. The symmetric difference quotient equation can be seen below.

$$\left.\frac{df(x)}{dh}\right|_{x=h_0} \approx \frac{f(h_0 + \Delta)) - f(h_0 - \Delta))}{2 \cdot \Delta} \tag{88}$$

Using equation (*88*), it is possible to generalize the partial derivatives of the vector $\vec{\boldsymbol{p}}$, by denoting indexation with the operator []. That means, the component j of the vector $\vec{\boldsymbol{p}}$ is denoted as $\vec{\boldsymbol{p}}[j]$. The function $\vec{\boldsymbol{\Delta}}(i)$ is defined in equation (*76*) such as $\vec{\boldsymbol{p}} + \vec{\boldsymbol{\Delta}}(2 \cdot j - 1)$ represents a negative Δ change and $\vec{\boldsymbol{p}} + \vec{\boldsymbol{\Delta}}(2 \cdot j)$ represents a positive Δ change in the component j of the vector $\vec{\boldsymbol{p}}$. Using the same considerations stated in section 3.5.3, any partial derivative of the cost function $f_c(\vec{\boldsymbol{p}})$ is calculated as follows:

$$\left.\frac{\partial f_c(\vec{\boldsymbol{p}})}{\partial \vec{\boldsymbol{p}}[j]}\right|_{\vec{\boldsymbol{p}}=\vec{\boldsymbol{p}}_0^{(k)}} \approx \frac{f_c\left(\vec{\boldsymbol{p}}^{(k)} + \vec{\boldsymbol{\Delta}}(2 \cdot j)\right) - f_c\left(\vec{\boldsymbol{p}}^{(k)} + \vec{\boldsymbol{\Delta}}(2 \cdot j - 1)\right)}{2 \cdot \Delta} \;;\; j \in [1, \ldots, 6] \tag{89}$$

Equation (*89*) indicates that the partial derivative of the component j is created using the image similarity measurements calculated by the fixed X-ray image and the two DRR images created by changing the component j with a positive and a negative Δ. The obtained gradient is given to the optimization function, which computes a new pose to be evaluated.

5.2.2. Finding the C-arm Coordinate System

Based on the C-arm mathematical model in section 4.5.3, a real C-arm and a C-arm represented by the DRR will generate a similar image if the patient's (or image modality) position with respect to the C-arm (or virtual C-arm) is approximately the same. The objective of this section is to find the connection between the virtual and real C-arm reference frames.

It is then necessary to build the reference frame of the used C-arm, Ziehm Vario 3D, using the mathematical model in section 4.5.3. This reference frame is constructed experimentally by taking an X-ray of an object, defining X and Y coordinates in the image, i.e., finding the X-ray reference frame on the image. After that, the corresponding axes in the setup can be obtained, i.e., finding the retinal plane of the C-arm reference frame. On the left side of Figure 5-5, it can be seen the X-ray of an RB. The X coordinate is defined as the first element of the tuple defining the pixel position on the image, and the Y coordinate defines the second element of the tuple. The green coordinate system (on the left side of Figure 5-5) is drawn, such as the previous definition is fulfilled. Then the retinal plane of the C-arm is built consistent with the X-ray reference frame. That means the retinal plane axes match the X-ray reference frame as the right image in Figure 5-5 shows.

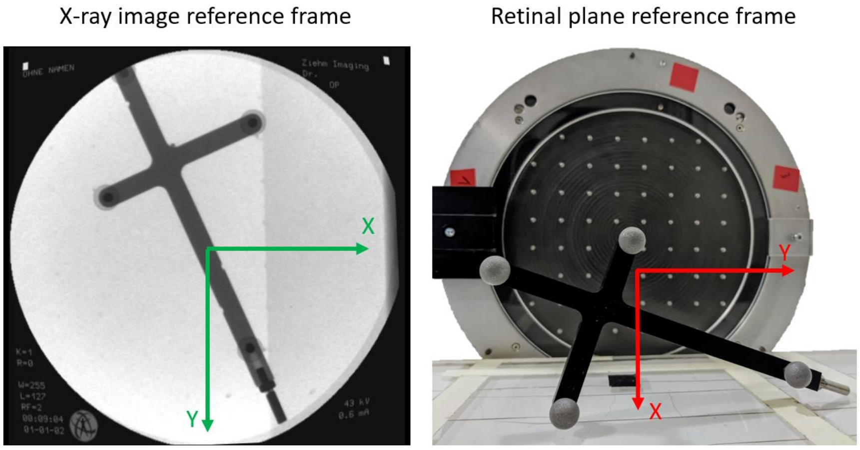

Figure 5-5. Experimental definition of the X-ray reference frame and the retinal plane reference frame

Once the retinal plane is built, it can be extended to create the C-arm reference frame. By definition of the C-arm mathematical model in section 4.5.3, the retinal plane and the focal plane, formed by the X-Y plane of the C-arm reference frame, are distanced by the focal length in the Z-coordinate without any rotation. Bearing that, the {C-ARM} frame can be found as Figure 5-6 shows.

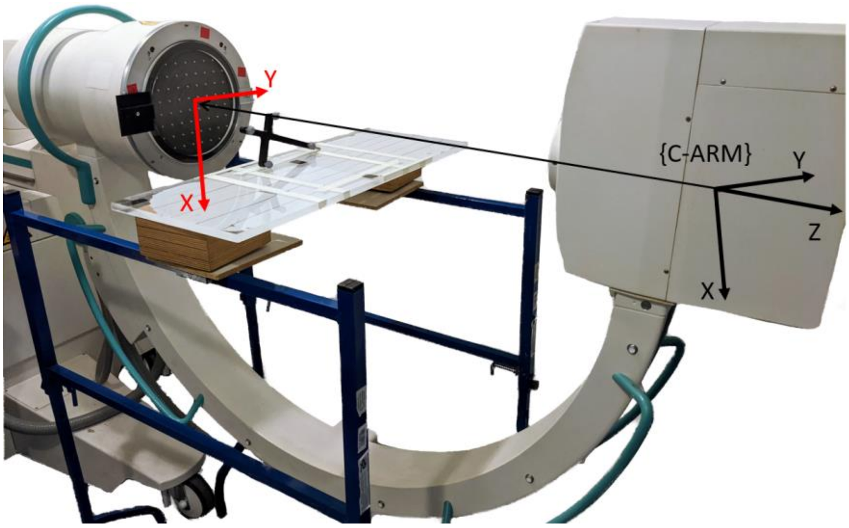

Figure 5-6. Experimental definition of the C-arm reference frame, {C-ARM}

5.2.3. Finding the Transformation from the DRR to the C-arm

Now that {C-ARM} is contextualized with respect to the physical C-arm and the DRR reference frame is known (see Figure 4-23), the next step is to find the transformation between the two frames. In Figure 5-7, the reference frames of the C-arm and the DRR are shown side by side. The DRR image and the X-ray image are compared during the registration process, so the connection between the DRR and the C-arm is created through the center of the images.

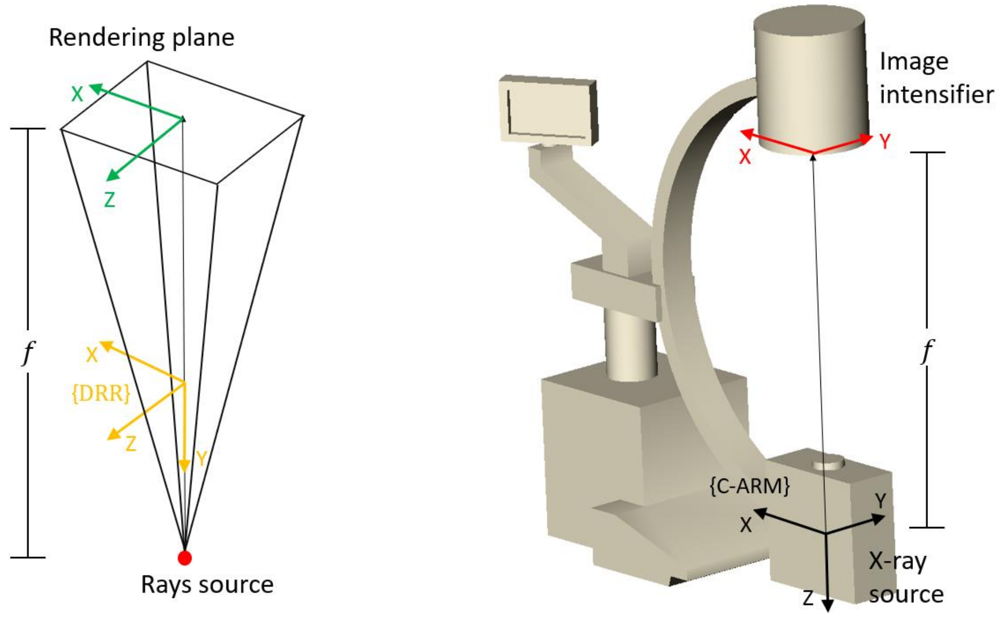

Figure 5-7. Comparison of the reference frames of virtual C-arm (DRR) on the left side and a real C-arm on the right

The transformation from the virtual to the real C-arm can be calculated, from Figure 5-7, using the transformations expressed in equation (*90*).

$$^{DRR}\boldsymbol{T}_{C-ARM} = {}^{DRR}\boldsymbol{T}_{RenderingPlane} \cdot {}^{RenderingPlane}\boldsymbol{T}_{ImageIntensifier} \cdot {}^{ImageIntensifier}\boldsymbol{T}_{C-ARM} \quad (90)$$

The transformation $^{DRR}\boldsymbol{T}_{RenderingPlane}$ is expressed as a translation of $-f/2$ on the Y-axis. The rotation between the DRR render frame and the X-ray image frame is represented by $^{RenderingPlane}\boldsymbol{T}_{ImageIntensifier}$. The DRR rendering is explained in section 3.5.1, and the X-ray image frame is explained above. To compare both images, their origins are aligned to be the same, so this transformation is reduced to a negative 90° rotation in the X-axis. The transformation $^{ImageIntensifier}\boldsymbol{T}_{Carm}$ is a positive f displacement in the Z-axis. With the above considerations, equation (*90*) can be evaluated as:

$$^{DRR}\boldsymbol{T}_{C-ARM} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & -f/2 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & -1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & f \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (91)$$

Simplifying (*91*), $^{DRR}\boldsymbol{T}_{C-ARM}$ is found to be:

$$^{DRR}\boldsymbol{T}_{C-ARM} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & f/2 \\ 0 & -1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (92)$$

The same C-arm device is used during the development of this work, and the mathematical conventions used for the DRR and the C-arm model does not change. Consequently, $^{DRR}\boldsymbol{T}_{C-ARM}$ remains constant.

5.2.4. Transforming the 2D/3D Registration Result into a Useful Frame

During the registration process, the optimization parameter, as shown in equation (*47*), is $\vec{\boldsymbol{p}}$. The output of the registration is the optimized transformation $\vec{\boldsymbol{p}}_R$, which can also be expressed as ${}^{DRR}\boldsymbol{T}_{DICOM}$. In a graphical interpretation, from equation (*86*), $\vec{\boldsymbol{p}}_R$ is the pose that makes the DRR image look most alike with respect to the X-ray image. Although the optimized ${}^{DRR}\boldsymbol{T}_{DICOM}$ is explained with the previous graphical interpretation, it does not give useful information as the used frames are abstract, i.e., {DRR} and {DICOM}.

Therefore, it is meaningful to transform the registration result into a physical reference frame. The transformation obtained from the registration result uses reference frames given by the camera feedback, which are not used during the registration process. Later, it will be seen that the camera feedback is not used for the registration, in the case when only one X-ray image is used.

Making use of the equation (*87*), the registration results ${}^{DRR}\boldsymbol{T}_{DICOM}$ can be transformed into ${}^{ARB}\boldsymbol{T}_{DICOM}$ using the camera feedback, i.e., ${}^{OTS}\boldsymbol{T}_{CRB}$ and ${}^{OTS}\boldsymbol{T}_{ARB}$, and the transformation from the virtual to the real C-arm, ${}^{DRR}\boldsymbol{T}_{C-ARM}$. The registration result in ${}^{ARB}\boldsymbol{T}_{DICOM}$ brings the advantage to be already on a reference frame that a robotic surgical system can use as a target, see section 5.1.2. In addition, since ARB is the rigid body attached to the patient, it keeps the registration aligned with the patient regardless of any patient movement. That means the DICOM reference frame is bound to {ARB}. In other words, as long as ARB is fixed to the patient, any patient movement is inherently compensated by having the registration on this frame. Guaranteeing the attachment of ARB to a patient is out of the scope of this work. In the used phantoms, ARB is attached to a rigid acrylic structure using screws, but in an actual surgical scenario, keeping an RB fixed to a patient is a complex topic still under development [155].

5.2.5. Introduction to the Region of Interest (ROI) Selection for the Registration Process

The size of the X-ray images used for the registration is 568x568 pixels, as this is the size of the Ziehm Vario 3D output image. However, the C-arm raw images contain some labels on the image edges, which are not important for the 2D/3D registration. A part from that, the entire body segment captured by the X-ray may not be relevant for the registration. For example, an X-ray image captures four vertebrae plus some surrounding soft-tissue, but it is desired to register a specific vertebra. It is computationally cheaper and procedurally more meaningful to reduce the area of the image that is used for the registration. The specific part of the image that the user can define is called the region of interest (ROI). Using the definition of an image proposed in *(52)*, the image domain is defined by the set $\mathbf{T}$. Let two points in the image, $\boldsymbol{c}_1$ and $\boldsymbol{c}_2$, be define as

$$\{(x_1, y_1), (x_2, y_2) \in \mathbf{T} \mid \boldsymbol{c}_1 = (x_1, y_1), \boldsymbol{c}_2 = (x_2, y_2)\} \tag{93}$$

The ROI is a square defined by points $\boldsymbol{c}_1$ and $\boldsymbol{c}_2$, representing the end-points of the square diagonal.

Mathematically the ROI can be described as a function that receives an image and two points and returns the sub-image inscribed by the squared defined by points, $\boldsymbol{c}_1$ and $\boldsymbol{c}_2$.

$$\boldsymbol{I}_{ROI} = f_{ROI}(\boldsymbol{I}_{Xray}, \boldsymbol{c}_1, \boldsymbol{c}_2) \tag{94}$$

An implicit property following from equation (*93*) indicates that the image dimension after the ROI process is at most equal to the input image. Nevertheless, the idea behind the ROI operation is to focus on a specific part of the input image, i.e., the output image size is generally smaller than the input image size. From $\boldsymbol{c}_1$ and $\boldsymbol{c}_2$, it is possible to compute the area of the ROI as:

$$A_{ROI} = |x_1 - x_2| \cdot |y_1 - y_2| \tag{95}$$

It is worth mentioning that the ROI selection is made on the X-ray image. The coordinates of the ROI are later on included in the DRR module, which renders the DRR image only within the selected ROI, allowing the DRR generation to be faster. In addition, only the area of the X-ray and DRR images defined by the ROI is used in the calculation of the similarity measurement. It makes the similarity measurement calculation faster, and it only compares the part of the images defined by the ROI.

5.2.6. Synchronization of AP and LAT Images in the Registration

Until this point, the registration procedure has considered only the utilization of one X-ray image, but the possibility of using N X-ray images has been stated. Since N images are aimed to calculate one registration, it is necessary to modify the current procedure to accept more images. The modification requires synchronizing the generation of DRR images followed by averaging the calculation of every similarity measurement. The synchronization requires the C-arm poses with respect to the navigation system in the moment of acquiring each of the N X-ray images. These poses are used to calculate the C-arm transformation from position A to position B, i.e., ${}^{C-ARM_\mathrm{A}}\boldsymbol{T}_{C-ARM_\mathrm{B}}$. See Figure 5-8.

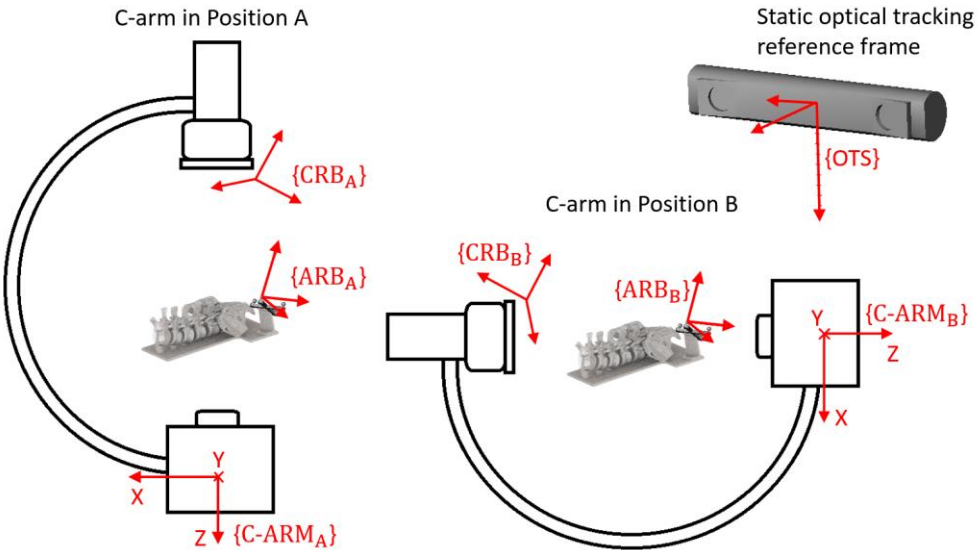

Figure 5-8. Reference frame in order to synchronize two X-ray images

$$ {}^{C-ARM_A}\boldsymbol{T}_{C-ARM_B} = {}^{C-ARM_A}\boldsymbol{T}_{CRB_A} \cdot {}^{CRB_A}\boldsymbol{T}_{OTS} \cdot {}^{OTS}\boldsymbol{T}_{CRB_B} \cdot {}^{CRB_B}\boldsymbol{T}_{C-ARM_B} \quad (96) $$

Where $\left({}^{CRB_A}\boldsymbol{T}_{C-ARM_A}\right)^{-1}$ and ${}^{CRB_B}\boldsymbol{T}_{C-ARM_B}$ are found with the C-arm calibration procedure explained in section 3.3.4, and $({}^{OTS}\boldsymbol{T}_{CRB_A})^{-1}$ and ${}^{OTS}\boldsymbol{T}_{CRB_B}$ are giving by the optical tracking system. Then equation (*96*) is used to find the transformation from the DRR representing the C-arm position A to the DRR representing the C-arm position B, ${}^{DRR_A}\boldsymbol{T}_{DRR_B}$. It goes as follows:

$$ {}^{DRR_A}\boldsymbol{T}_{DRR_B} = {}^{DRR_A}\boldsymbol{T}_{C-ARM_A} \cdot {}^{C-ARM_A}\boldsymbol{T}_{C-ARM_B} \cdot {}^{C-ARM_B}\boldsymbol{T}_{DRR_B} \quad (97) $$

As the same C-arm is used for both images, then the same mathematical model, ${}^{DRR}\boldsymbol{T}_{C-ARM}$, can be used in position A and B. It means, from equation (*92*), ${}^{DRR_A}\boldsymbol{T}_{C-ARM_A} = {}^{DRR_B}\boldsymbol{T}_{C-ARM_B} = {}^{DRR}\boldsymbol{T}_{C-ARM}$.
Using equation (*97*), the equation of the synchronization can be found, which relates the optimizing transformation in image A, ${}^{DRR_A}\boldsymbol{T}_{DICOM}$, to the optimizing transformation in image B, ${}^{DRR_B}\boldsymbol{T}_{DICOM}$:

$$ {}^{DRR_B}\boldsymbol{T}_{DICOM} = ({}^{DRR_A}\boldsymbol{T}_{DRR_B})^{-1} \cdot {}^{DRR_A}\boldsymbol{T}_{DICOM} \quad (98) $$

The equation of the synchronization, (*98*), automatically generates the changes in the DRR module B while the optimizer keeps iterating with respect to transformations in the DRR module A. The DRR module A and module B transformations can be represented as the vectors $\vec{\boldsymbol{p}}_A$ and $\vec{\boldsymbol{p}}_B$, respectively. The X-ray image A is compared with the DRR image A and the X-ray image B is compares with the DRR image B. The output of both similarity measurements is weighted averaged. The weight used for the average is the area of the ROI used in the respective X-ray image. The cost function, initially expressed in (*73*), is updated for two images, as follows:

$$ f_{c_{2\_images}}\left(\boldsymbol{I}_{ROI_A}, \boldsymbol{I}_{ROI_B}, \boldsymbol{V}_{CTscan}, \vec{\boldsymbol{p}}_A, \vec{\boldsymbol{p}}_B, A_{ROI_A}, A_{ROI_B}\right) = \frac{A_{ROI_A} \cdot f_c\left(\boldsymbol{I}_{ROI_A}, \boldsymbol{V}_{CTscan}, \vec{\boldsymbol{p}}_A\right) + A_{ROI_B} \cdot f_c\left(\boldsymbol{I}_{ROI_B}, \boldsymbol{V}_{CTscan}, \vec{\boldsymbol{p}}_B\right)}{2 \cdot \left(A_{ROI_A} + A_{ROI_B}\right)} \quad (99) $$

Equation (*99*) can be further generalized to the N images case, as equation (*100*) shows. It can be noticed that the equation (*97*) must be used N times to find from ${}^{DRR_A}\boldsymbol{T}_{DRR_B}$ until ${}^{DRR_A}\boldsymbol{T}_{DRR_N}$. Then, the N DRR modules will create N DRR images that will be compared N times with the N X-ray images using N image similarity measurements using the ROI of each of the N images.

$$ f_{c_{N\_Images}}\left(\boldsymbol{I}_{ROI_A}, \ldots, \boldsymbol{I}_{ROI_N}, \boldsymbol{V}_{CTscan}, \vec{\boldsymbol{p}}_A, \ldots, \vec{\boldsymbol{p}}_N, A_{ROI_A}, \ldots, A_{ROI_N}\right) = \frac{\sum_{i=A}^{N} A_{ROI_i} \cdot f_c\left(\boldsymbol{I}_{ROI_i}, \boldsymbol{V}_{CTscan}, \vec{\boldsymbol{p}}_i\right)}{N \cdot \sum_{i=A}^{N} A_{ROI_i}} \quad (100) $$

Regardless of the number of used images for the registration, the optimizer will receive the same gradient as in equation (*89*), and the pose to be optimized is ${}^{DRR_A}\boldsymbol{T}_{DICOM}$. This ${}^{DRR_A}\boldsymbol{T}_{DICOM}$ can be converted to any of the ${}^{DRR_N}\boldsymbol{T}_{DICOM}$, and using the camera feedback for that specific C-arm position, e.g. ${}^{OTS}\boldsymbol{T}_{ARB_N}$ and ${}^{OTS}\boldsymbol{T}_{CRB_N}$, will be transferred to ${}^{ARB}\boldsymbol{T}_{DICOM}$ using the equation *(87)*.

5.2.7. Evaluating the 2D/3D registration accuracy

Once the registration procedure is executed, there is no direct indication about the registration accuracy. Another laboratory experiment can be used to determine the registration accuracy indirectly. As requirements for the 2D/3D registration procedure, a CT-scan and a navigation system are required. Instead of using the robot system, a pointer that the navigation system tracked, is used for the experiment. As the pointer and testing device pose can be recorded, the following evaluation process can be automated. It means more samples can be analyzed to reach a better estimation of the 2D/3D registration accuracy, and only using the navigation system avoids the risk that the robot introduces some errors.

For the experiment, a testing device with an RB attached to it is required. The RB installed on the testing device is called DRB. The testing device must contain at least one well-defined landmark. This landmark must be known in the CT-scan, i.e., the point where the mark is located is uniquely distinguishable in the CT-scan. The landmark in the DICOM reference frame is expressed using the transformation ${}^{DICOM}\boldsymbol{T}_{Landmark}$. With the help of the pointer, the landmark is also determined in {OTS}, ${}^{OTS}\boldsymbol{T}_{Landmark}$, and transformed to {DRB}, i.e., ${}^{DRB}\boldsymbol{T}_{Landmark}$.

$$ {}^{DRB}\boldsymbol{T}_{Landmark} = {}^{TRB}\boldsymbol{T}_{OTS} \cdot {}^{OTS}\boldsymbol{T}_{Landmark} \tag{101} $$

The main conclusion from section 5.1.3 is to summarize the registration results as the transformation from {DICOM} to the physical frame attached to the element to be registered, the testing device in this case. Consequently, the result of registering the testing device with its CT-scan is ${}^{DRB}\boldsymbol{T}_{DICOM}$. The evaluation of the registration is done by transforming the landmark pose in DRB, ${}^{DRB}\boldsymbol{T}_{Landmark}$, to the DICOM frame, ${}^{DICOM}\boldsymbol{T}_{Landmark'}$

$$ {}^{DICOM}\boldsymbol{T}_{Landmark'} = ({}^{DRB}\boldsymbol{T}_{DICOM})^{-1} \cdot {}^{DRB}\boldsymbol{T}_{Landmark} \tag{102} $$

The translation and rotation of the same landmark due to the registration result is expressed as ${}^{Landmark'}\boldsymbol{T}_{Landmark}$. With the help of (*102*), ${}^{Landmark'}\boldsymbol{T}_{Landmark}$ is found as

$$ {}^{Landmark'}\boldsymbol{T}_{Landmark} = ({}^{DICOM}\boldsymbol{T}_{Landmark'})^{-1} \cdot {}^{DICOM}\boldsymbol{T}_{Landmark} \tag{103} $$

Consequently, ${}^{Landmark'}\boldsymbol{T}_{Landmark}$ is used as the metric for evaluating the accuracy of the 2D/3D registration. The translation vector that expresses the landmark movement due to the registration, i.e., ${}^{Landmark'}\boldsymbol{p}_{Landmark}$ is taken from the transformation ${}^{Landmark'}\boldsymbol{T}_{Landmark}$. The error of the registration due to the translation mismatch is computed as the squared root of ${}^{Landmark'}\boldsymbol{p}_{Landmark}$ magnitude, i.e. ℓ_2 norm.

$$ Reg\_error\_translation = \sqrt{({}^{Landmark'}\boldsymbol{p}_{Landmark})^{T} \cdot {}^{Landmark'}\boldsymbol{p}_{Landmark}} \tag{104} $$

An additional procedure is introduced to evaluate the rotation accuracy of the registration. There are two purposes behind this procedure: facilitating the comprehension of rotation mismatching and having both errors in the same units.

In this procedure, the landmark is assumed to be touched by the tip of a 40mm length pedicle screw, a standard pedicle screw size in spine surgery [156]. Let the origin of the screw coordinate frame be located on the screw tip, and the Z-axis be aligned to the pedicel screw length. The point containing the coordinate from screw tip to head is defined as ${}^{Landmark}\boldsymbol{p}_{screwHead}$. The displacement between the planned and the measured screw head, i.e., ${}^{screwHead}\boldsymbol{p}_{screwHead'}$ is calculated using the equation (*105*)

$$ {}^{screwHead}\boldsymbol{p}_{screwHead'} = {}^{screwHead}\boldsymbol{p}_{Landmark} + {}^{screwHead}\boldsymbol{R}_{Landmark} \cdot {}^{Landmark}\boldsymbol{R}_{Landmark''} \cdot {}^{Landmark''}\boldsymbol{p}_{screwHead'} \tag{105} $$

where ${}^{Landmark''}\boldsymbol{p}_{screwHead'} = [0 \quad 0 \quad -40]^T$, ${}^{screwHead}\boldsymbol{p}_{Landmark} = [0 \quad 0 \quad 40]^T$, and ${}^{Landmark}\boldsymbol{R}_{Landmark''} = {}^{Landmark}\boldsymbol{R}_{Landmark'}$,which was found in (*103*).

The X and Y components of ${}^{screwHead}\boldsymbol{p}_{screwHead'}$ represent the screw head displacement proportionally to the rotation error. In this way, the rotation accuracy is transformed into a translation. The point ${}^{screwHead}\boldsymbol{p}_{screwHead''}$ with the Z-component ignored, i.e., $z = 0$, is called ${}^{screwHeadXY}\boldsymbol{p}_{screwHead'}$. The error of the registration due to the angle mismatch is calculated using equation (*104*), but using ${}^{screwHeadXY}\boldsymbol{p}_{screwHead'}$ instead. The accuracy approach to calculate the registration error can be seen in Figure 5-9.

$$ Reg\_error\_rotation = \sqrt{({}^{screwHeadXY}\boldsymbol{p}_{screwHead'})^T \cdot {}^{screwHeadXY}\boldsymbol{p}_{screwHead'}} \tag{106} $$

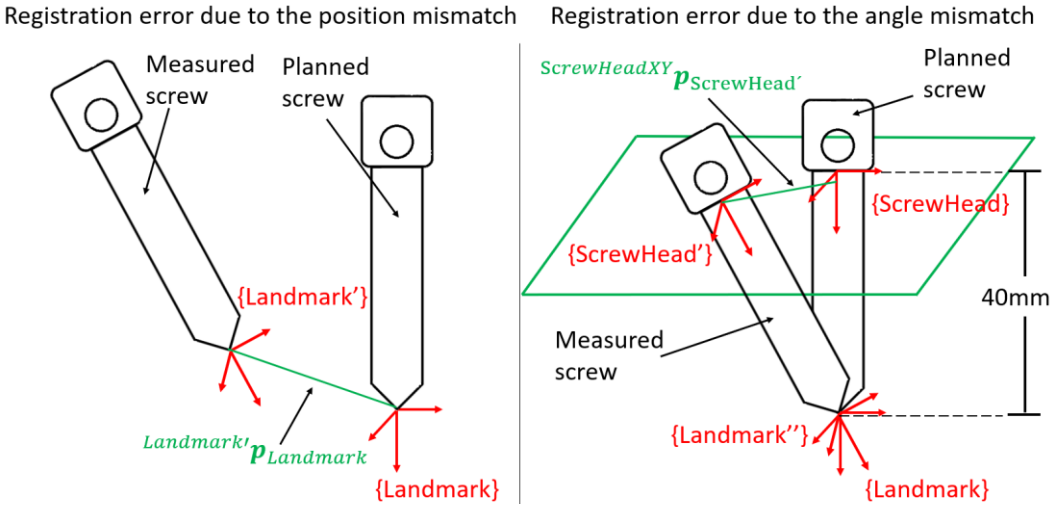

Figure 5-9. Rotation accuracy approach transformed into translation accuracy

The registration errors due to the translation and the rotation have the same units as the coordinates. In this work, the desired accuracy is in the order of the millimeters. It is meaningful to express the coordinates, and consequently, the registration error in millimeters.

The registration error due to the translation mismatch can be seen as the displacement of the implant tip. It can also be interpreted as the radius of the sphere that describes the error mismatch between the planned and measured screw.

The registration error due to the angle mismatch is the displacement of the head of the implants mismatch if the tips of the implants were together. The registration error due to the angle can also be seen as the maximum radius of the circle that describes an additional error mismatch. In that way, both errors can be combined into a single representation, as Figure 5-10 shows.

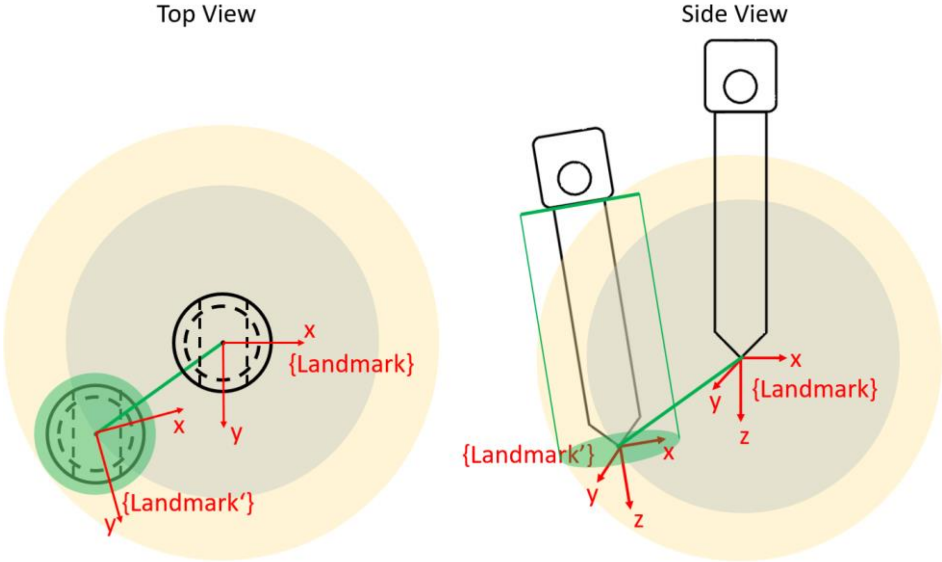

Figure 5-10. Total registration error

Mathematically, the maximum registration error can be found as the arithmetic addition of equations *(104)* and *(106)*, and the same expression can be used as the approximation of the registration error assuming small angle approximations. The last described calculations are expressed in equations (*107*) and (*108*):

$$Max\_Reg\_error = Reg\_error\_translation + Reg\_error\_rotation \tag{107}$$

$$Reg\_error \approx Max\_Reg\_error \tag{108}$$

Although the implant breach, equation (*29*), is the standard measure for implant quality, it is not used to assess the registration quality. The implant breach, explained in section 4.4, ignores the deviation along the implant axis [18], making the assessment of the registration quality inaccurate. The error calculation in equation (*108*) considers a 3D error plus the contribution of the orientation error. The implant breach is a simpler 1D error. Consequently, the error calculation in equation (*108*) can be seen as a general case of the implant breach calculation. It means the implant breach would be smaller or (at most) equal to the registration accuracy assessed using equation (*108*). If the implant error is smaller than 2mm using equation (*108*), it is possible to classify this implant with Grade A+B in the Gertzbein-Robbins scale.

5.2.8. Discussion and Conclusions

The numerical partial derivative of the cost function of a specific component is calculated by finding the image similarity measurement of the current pose plus a small offset and minus the same offset in the desired component. Once the two similarity measurements are found, the symmetric difference quotation equation is used. See equation *(89)*.

${}^{DRR}\boldsymbol{T}_{C-ARM}$ is calculated through a chain of transformations that has a common point, the DRR and X-ray image center. The calculated transformation remains constant as the same C-arm is used during this work, and the DRR and C-arm model does not change.

The registration result from the minimization procedure is the transformation ${}^{DRR}\boldsymbol{T}_{DICOM}$. This value does not present a meaningful result for a robot-assisted-surgical system. Using the surgical tracking system feedback, the registration result can be expressed as the meaningful ${}^{ARB}\boldsymbol{T}_{DICOM}$ transformation, which links the DICOM volume reference frame with the rigid body attached to the patient.

The region of interest selection presents a solution to optimize the computational power used during the registration and focus the registration procedure on a specific part of the X-ray image.

When more than two images are used to carry out the 2D/3D registration, it is necessary to know the transformation among images, e.g., ${}^{DRR_A}\boldsymbol{T}_{DRR_N}$. This transformation is the key to synchronizing the DRR image module A, ${}^{DRR_A}\boldsymbol{T}_{DICOM}$, with the DRR image module N, ${}^{DRR_N}\boldsymbol{T}_{DICOM}$. After each DRR image is rendered, N image similarity measurements are calculated between the N X-ray images and the N DRR images, respectively. The weighted average of the similarity measurements is computed using the corresponding ROI area of each image as the respective weight. The similarity measurements weighted-average is used as the input for the optimizer as in the 2D/3D registration with one X-ray image.

A procedure to automate the computation of the registration accuracy is developed. It uses a landmark on the testing subject as the registration accuracy cannot be directly measured from the registration result, i.e., ${}^{ARB}\boldsymbol{T}_{DICOM}$. It mixes the error due to position and the angle mismatch and gives the total error in millimeter units.

5.3. Undistortion of X-Ray Images from Conventional Panel C-arm

In conventional C-arms, distortions presented in X-ray images are a mixture of barrel, S-shape, and spiral distortions [26] [27]. One of the biggest causes for the distortions is the earth's magnetic field, which affects the trajectory of the electrons inside the image intensifier depending on the C-arm pose [28]. As this distortion is time, position, and orientation-dependent, it is hardly possible to create a general model of the C-arm distortion.

C-arm images with distortion are widely accepted and used in general surgery due to their versatility and low-cost [157]. However, there are two reasons to use images without distortion in the 2D-3D registration. The first reason, the C-arm parametrization, explain in sections 4.5.4 and 5.4, uses a relation of 3D and 2D positions of a set of fiducials. The set of 2D fiducial positions is extracted from an X-ray image. Miscalculations in detecting the centers of the fiducials lead to misestimating the C-arm parameters, and consequently, the 2D/3D registration result. The second reason, the registration process compares an X-ray image with a DRR image, which is intrinsically distortion-free. The comparison is consistent if the nature of both images is the same.

In this section, an undistortion approach on C-arm images using a plate with fiducials in a grid shape is explained [158]. A fiducial detection method is developed and applied to warp X-ray images [159]. After the undistortion is carried out, the fiducials are deleted from the image using an inpainting algorithm. The inpainting process is incorporated to improve body segment visibility.

Although the undistortion plate is fixed on the C-arm detector and aligned using C-arm rigid points given by the manufacturer, an analysis of the positions of the fiducials in the X-ray images with and without distortion shows a translation between image centers and a rotation between images. The results of the analysis are used to calculate a rigid transformation from the original X-ray image to the image without distortion. Incorporating this rigid transformation after the image undistortion process has not been previously reported in the literature.

5.3.1. Mapping Algorithm for X-Ray Images Undistortion

As explained before, X-ray images from conventional C-arm suffer several types of distortion, such as barrel, S-shape, and spiral distortion. Since spiral distortions are influenced by the earth's magnetic field, which is variable, it is hardly possible to find a general undistortion map describing distortions of future C-arm images. Instead, it is common to attach an undistortion device to the C-arm detector, which generates an undistortion map for each X-ray image. The undistortion device contains fiducials in a known layout. This layout is used to generate a mapping function between the currently known image with distortions and the future image without distortion. The mapping function is created using a bi-polynomial regression, and the creation of the image without distortion is done by an inverse warping procedure using bilinear interpolation [158] [160] [161].

Mapping Function Creation

Let $(x_d, y_d) \in \mathbb{R}^2$ and $(x_u, y_u) \in \mathbb{R}^2$ be a correspondent point in an image with and without distortion, respectively. The function mapping a point from the image without distortion to the original image, $f_{map}(x_u, y_u)$, is defined as a bi-polynomial regression function. Let K and L be the polynomial degree of the variable x_u and y_u, respectively. The previous definitions are defined in (*109*) as

$$(x_d, y_d) = f_{map}(x_u, y_u): (x_u, y_u) \to (x_d, y_d)$$

$$x_d(x_u, y_u) = \sum_{k=0}^{K} \sum_{l=0}^{L} a_{kl} \cdot x_u^k \cdot y_u^l \tag{109}$$

$$y_d(x_u, y_u) = \sum_{k=0}^{K}\sum_{l=0}^{L} b_{kl} \cdot x_u^k \cdot y_u^l$$

where a_{kl} and b_{kl} are the coefficients of the bi-polynomial function for the coordinate x_d and y_d, respectively. For the particular case $K = L = 2$, the mapping function has the following matrix representation:

$$\begin{bmatrix} x_d \\ y_d \end{bmatrix}^T = [1 \quad x_u \quad y_u \quad y_u \cdot x_u \quad x_u^2 \quad y_u \cdot x_u^2 \quad y_u^2 \quad y_u^2 \cdot x_u \quad y_u^2 \cdot x_u^2] \cdot \begin{bmatrix} a_{00} & b_{00} \\ a_{10} & b_{10} \\ a_{01} & b_{01} \\ a_{11} & b_{11} \\ a_{20} & b_{20} \\ a_{21} & b_{21} \\ a_{12} & b_{12} \\ a_{02} & b_{02} \\ a_{22} & b_{22} \end{bmatrix} \tag{110}$$

Let $\vec{\boldsymbol{a}}_a$ be the column vector formed by the a_{kl} coefficients of the bi-polynomial function, $\vec{\boldsymbol{a}}_b$ be the column vector formed by the b_{kl} coefficients of the bi-polynomial function, and $\vec{\boldsymbol{x}}_c^{\mathrm{T}}$ be the row vector composed by the evaluation of a point in the image without distortion following the bi-polynomial function. From equation (*110*), the convention will be taken as follows:

$$\vec{\boldsymbol{a}}_a = [a_{00} \quad a_{10} \quad a_{01} \quad a_{11} \quad a_{20} \quad a_{21} \quad a_{12} \quad a_{02} \quad a_{22}]^T \tag{111}$$

$$\vec{\boldsymbol{a}}_b = [b_{00} \quad b_{10} \quad b_{01} \quad b_{11} \quad b_{20} \quad b_{21} \quad b_{12} \quad b_{02} \quad b_{22}]^T \tag{112}$$

$$\vec{\boldsymbol{x}}_c^T = [1 \quad x_u \quad y_u \quad y_u \cdot x_u \quad x_u^2 \quad y_u \cdot x_u^2 \quad y_u^2 \quad y_u^2 \cdot x_u \quad y_u^2 \cdot x_u^2] \tag{113}$$

Replacing the adopted convention described in (*111*), (*112*), and (*113*) in (*110*):

$$\begin{bmatrix} x_d \\ y_d \end{bmatrix}^T = \vec{\boldsymbol{x}}_c^T \cdot [\vec{\boldsymbol{a}}_a \quad \vec{\boldsymbol{a}}_b] \tag{114}$$

Equation (*114*) can be split in two equations as follows:

$$x_d = \vec{\boldsymbol{x}}_c^T \cdot \vec{\boldsymbol{a}}_a \tag{115}$$

$$y_d = \vec{\boldsymbol{x}}_c^T \cdot \vec{\boldsymbol{a}}_b \tag{116}$$

A calibration device with N fiducials gives $2 \times N$ corresponding points (fiducials), i.e., N points in the original image with distortion and N points in the image without distortion. The N points in the image with distortion (x_{d1}, y_{d1}), …, (x_{dN}, y_{dN}), are obtained from the X-ray image. The N points in the image without distortion (x_{u1}, y_{u1}), …,(x_{uN}, y_{uN}), are obtained from the design of the undistortion device design (see section 5.3.2). Let $\vec{\boldsymbol{x}}_d$ be a column vector with the X-coordinate of the N points in the image without distortion, $\vec{\boldsymbol{y}}_d$ be a column vector with the Y-coordinate of the N points in the image without distortion, and $\boldsymbol{A}_x$ a matrix that has in its k^{th}-row the respective $\vec{\boldsymbol{x}}_c^{\mathrm{T}}$ vector done with the k^{th} point of the image with distortion, i.e., (x_{dk}, y_{dk}). With the previous definitions and considering that the $2 \times N$ points are known, equations (*115*) and (*116*) are rewritten as:

$$\vec{\boldsymbol{x}}_d = \boldsymbol{A}_x \cdot \vec{\boldsymbol{a}}_a \tag{117}$$

$$\vec{\boldsymbol{y}}_d = \boldsymbol{A}_x \cdot \vec{\boldsymbol{a}}_b \tag{118}$$

Equations (*117*) and (*118*) are used to compute $\vec{\boldsymbol{a}}_a$ and $\vec{\boldsymbol{a}}_b$, containing the coefficients of the by-polynomial function, i.e., a_{kl} and b_{kl}. Assuming that there are more points than bi-polynomial coefficients, i.e., $\boldsymbol{A}_x \in \mathbb{R}^{m \times n}$ with $m > n$, the equations can be solved using the least squares solution (see Appendix A.5).

For this work, $K = L = 5$ as some previous studies have shown that a fifth order bi-polynomial manages to represent the image distortion on the C-arm properly [158] [160].

Rendering the Image Without Distortion using Image Warping

After the coefficients of the bi-polynomial function are known, the image without distortion is filled with the pixel information from the image with distortion. This last image creation process is called image warping. Let $\boldsymbol{I}_d(x_d, y_d)$ and $\boldsymbol{I}_u(x_u, y_u)$ be the images with and without distortion, respectively, which follow the image definition from (*51*). The image warping process is mathematically defined using the equation (*109*) as:

$$\boldsymbol{I}_u(x_u, y_u) = \boldsymbol{I}_d(f_{map}(x_u, y_u)) \tag{119}$$

The warping process is achieved once each pixel of $\boldsymbol{I}_u(x_u, y_u)$ corresponds with a location in $\boldsymbol{I}_d(x_d, y_d)$, as shown in Figure 5-11.

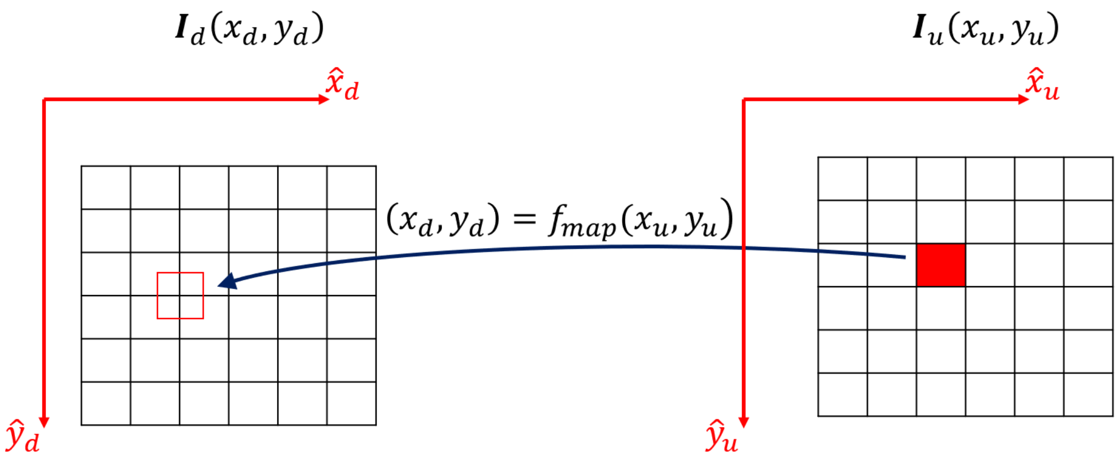

Figure 5-11. Representation of the image warping process

One problem arises when finding the pixels correspondences (x_u, y_u), using the bi-polynomial function (see equation (*109*)) as the obtained pixels (x_d, y_d), are real numbers, i.e., they land between pixels. By definition in (*51*), image pixels belong to the integer set of numbers, so the intensity level of a sub-pixel position is mathematically undetermined. A bilinear interpolation with the four closest pixels is used to estimate the intensity level of a sub-pixel. Figure 5-12 illustrates the sub-pixel problem and the used bilinear interpolation approach.

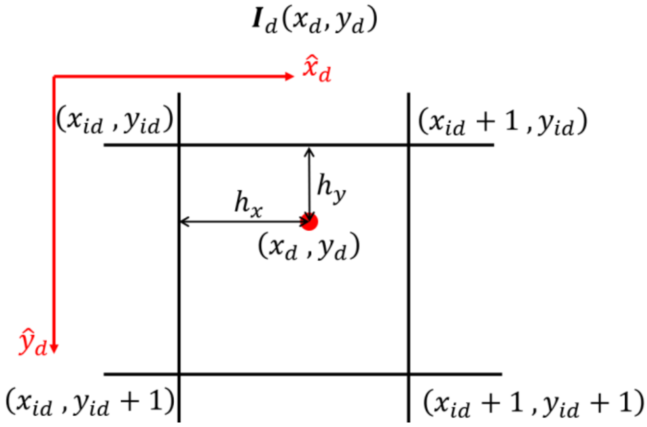

Figure 5-12. Interpolation of a sub-pixel with the surrounding pixels

Let (x_{id}, y_{id}) be the subpixel (x_d, y_d) point after applying the floor function, i.e., (x_{id}, y_{id}) is the lowest closest pixel to the subpixel (x_d, y_d). Let h_x and h_y be defined as the distances from the subpixel (x_d, y_d) to the left and upper edge of the square defined by the four closest pixels, respectively, i.e., $h_x = x_d - x_{id}$ and $h_y = y_d - y_{id}$. The pixel $(x_{id}+1, y_{id}+1)$ is the next in the diagonal movement with respect to (x_{id}, y_{id}). These last two pixels define a square of side 1px and area 1px^2, which inscribes the subpixel (x_d, y_d). This unit square is further divided into four squares. Each of these squares is defined by the diagonal made with the four pixels closest to the subpixel (x_d, y_d). The interpolation of the intensity level for the subpixel (x_d, y_d) is defined as the weighted addition of the intensity levels of surrounding pixels. The weights are the areas of the opposite square. The interpolation is mathematically defined with the following expression:

$$\boldsymbol{I}_d(x_d, y_d) \approx (1-h_x)\cdot(1-h_y)\cdot\boldsymbol{I}_d(x_{id}, y_{id}) + h_x\cdot(1-h_y)\cdot\boldsymbol{I}_d(x_{id}+1, y_{id}) + h_x\cdot h_y\cdot\boldsymbol{I}_d(x_{id}+1, y_{id}+1) + (1-h_x)\cdot h_y\cdot\boldsymbol{I}_d(x_{id}, y_{id}+1) \quad (120)$$

5.3.2. Design of the Undistortion Device for the X-Ray Images

The undistortion device is designed with a set of fiducials forming a grid, which is attached to the C-arm detector as the literature suggests [158] [162] [161] [160]. There are two points to consider for the undistortion device, grid size, and the number of fiducials in the grid. It is a common practice to spread the fiducials all around the image to obtain a better regression function. On the other hand, there is no agreement regarding the number of used fiducials. In one study, 300 fiducials were used in the undistortion device [158], 120 fiducials in another research [162], 77 in another [161], and 37 fiducials in another undistortion device [160]. In this work, the grid pattern is made as a symmetrical array of 8x8 fiducials. A black circular shape masks the X-ray images from the C-arm, so the corners of the grid are removed to maximize the cover by the fiducials in the image. The pattern design is composed of a grid of 60 fiducials, which lies between the two analyzed studies with the minimum number of fiducials.

The device is manufactured in acrylic using steel beads of 3mm in diameter, which enter in contact with the C-arm detector surface. The position of the fiducials in the image without distortion are taken by the physical grid dimensions of the device. The distance row- and column-wise between fiducials is 21.9mm. The pixel spacing of the C-arm detector gives the conversion from millimeters to pixels, 0.365mm/px for the used Ziehm Vario 3D. The selected distance between fiducials is a multiple of the pixel spacing parameter, making each fiducial position in the image without distortion an integer number.

For mounting, the undistortion device considers two threads available in the C-arm detector, which aligns the undistortion device with the detector. The computer-aided design (CAD) and the undistortion device attached to the C-arm detector are shown in Figure 5-13.

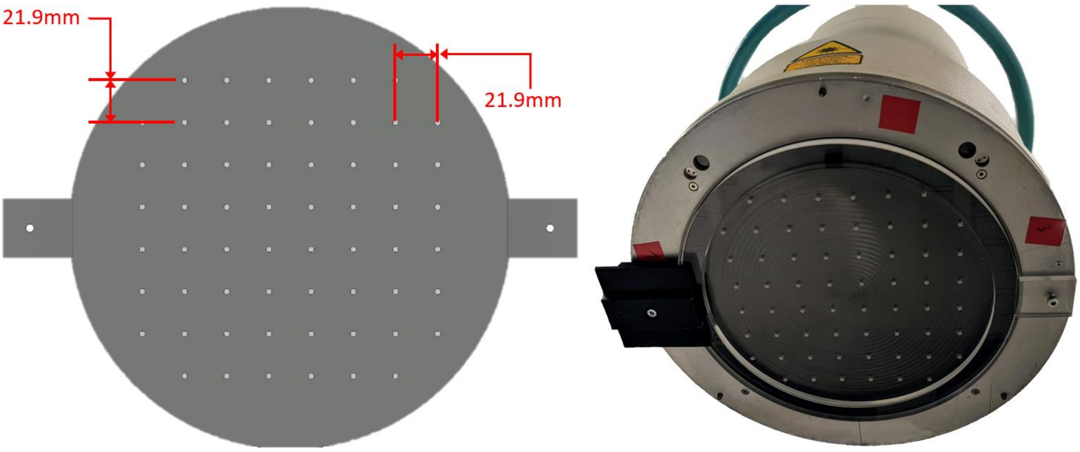

Figure 5-13. Undistortion device. CAD on the left. Manufactured and installed device on the right

5.3.3. X-Ray Image Undistortion Procedure and Considerations

The undistortion procedure is implemented by combining the warping algorithm and the fiducial detection of the undistortion device. A test to examine the undistortion procedure is executed using a brand-new K-wire, guaranteed to be straight. In Figure 5-14, it can be seen a brand-new K-wire, which follows the edge of a metric rule.

Figure 5-14. K-wire aligned with a rule

After an X-ray of the brand-new K-wire is taken, the distortion effects can be noticed with bare eyes, left side in Figure 5-15. Keeping the setup static, the undistortion device is attached to the C-arm detector, and another X-ray image is taken, right side Figure 5-15. It can be seen close to the center of both images that the K-wire looks straight, but the distortion is quite noticeable near the edges of the images.

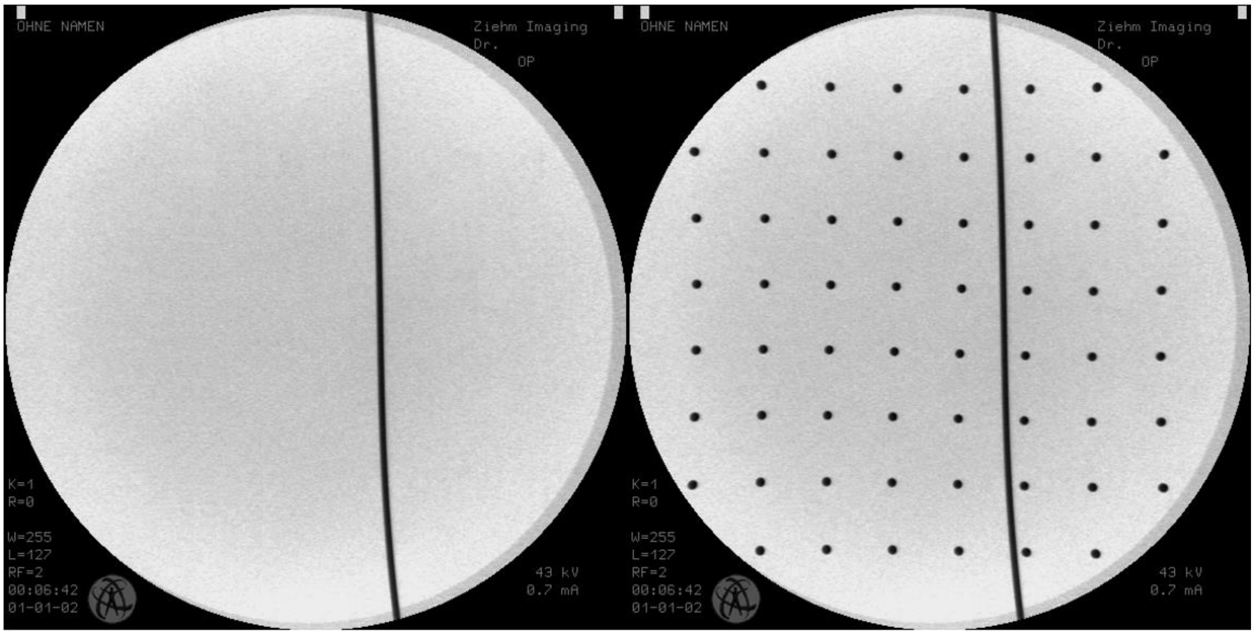

Figure 5-15. On the left side, X-ray of the K-wire. On the right side, same setup with the undistortion device

From the right side of Figure 5-15, the centers of the fiducials, projected from the distortion device, are calculated. The centers are selected manually at this stage, but a novel approach using deep-learning is described in section 7.1. These fiducials centers are used for building the vectors $\vec{\boldsymbol{x}}_d$ and $\vec{\boldsymbol{y}}_d$ of equations (*117*) and (*118*). Using the procedure described in section 5.3.1, the image can be mapped into an image without distortion, as the left side of Figure 5-16 shows.

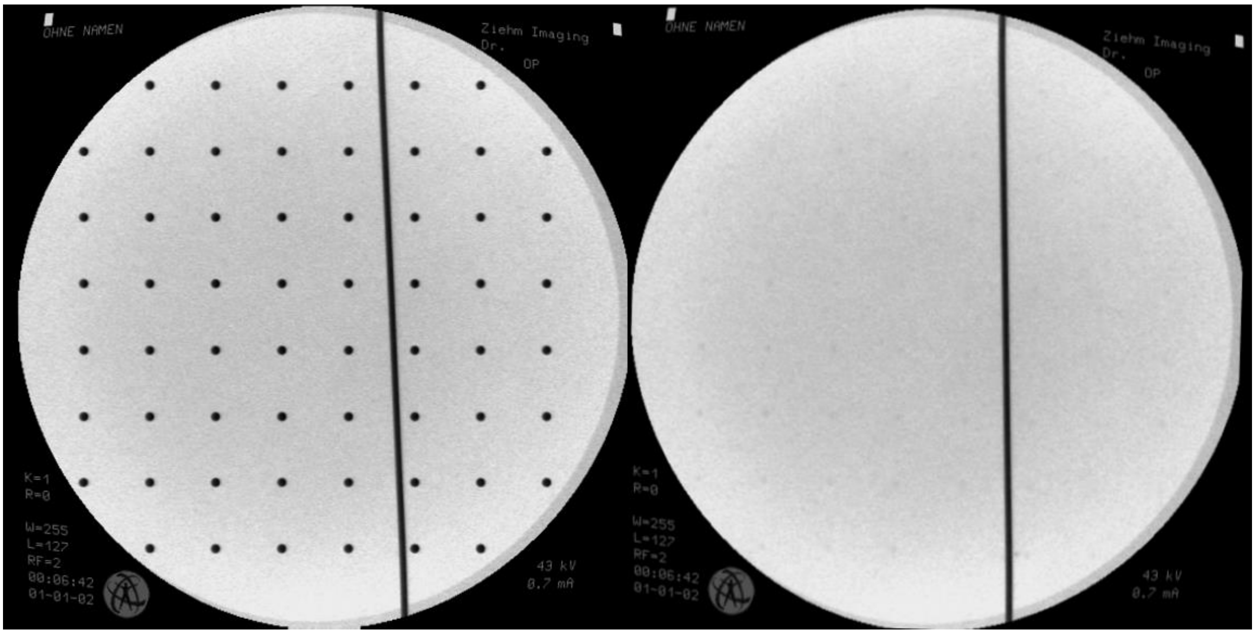

Figure 5-16. On the left side, image with corrected distortion. On the right side, image without distortion and inpainting

In section 5.3.3.1, it is explained the procedure used to remove the fiducials digitally from the image without distortion. It is known as *image inpainting*, and the result can be seen on the right side of Figure 5-16. For the rest of this section, the undistortion quality is examined.

The following test is carried out to illustrate the undistortion effectiveness; five X-rays of the K-wire are taken in different rotation and translations. Each image is processed to remove the distortions. The pixels forming the K-wire left edge are calculated and used to compute a linear regression, then the R-squared and the standard error of the regression are calculated (see Appendix A.7). The results can be seen in Table 5-1.

Table 5-1. Calculation of the standard error of the regression in X-ray images after distortion using a K-wire

Image Number	Original X-ray image		X-ray Image after undistortion	
	R^2	Standard error [px]	R^2	Standard error [px]
1	0.9996	2.4813	1	0.6661
2	0.9998	1.6758	1	0.6201
3	0.9179	41.2758	0.9984	5.6053
4	0.9997	1.7659	1	0.7476
5	0.9355	1.4416	0.9877	0.2481

As it can be noticed, the standard error of the regression on the images without distortion is lower than in the images with distortion. The R-squared values are closer to one in the tested cases after the undistortion process, which describes the fit of the K-wire points using a linear regression model. The standard error represents the absolute fit of the linear regression to the K-wire points. In the tested images, the standard error on the images without distortion shows with one exception that the error is smaller than 1 pixel. Although the R-squared and standard error of the regression do not match 100% of the K-wire points with a line, the fit improves compared with the original image.

Two possible reasons are explaining the obtained errors. The first reason, the edge detector introduces some errors as it uses a threshold operation before detecting the edges. The second reason, as the regression is done in $\mathbb{R}^2$, and the image pixels $\in \mathbb{Z}^2$; a quantization error impacts the standard error of the regression. A quantization error is a common phenomenon in every analog to digital conversion, and measuring the fit of a straight device after being digitalized brings this inherent problem that cannot be ignored.

5.3.3.1. Inpainting fiducials

Image inpainting is used for removing the fiducials out of the image after the undistortion process. This technique is commonly used to restore old and degraded photos, where scratches, black spots, and imperfections are present. The inpainting procedure is carried out by growing from the boundaries towards the center of a defined region. Every pixel in the boundary to be inpainted is replaced by a normalized weighted sum of the known neighbor pixels. The process is repeated until each pixel inside the region is processed [163]. As the undistortion process is performed using the ideal positions of the fiducials, then the positions of the fiducials after the undistortion are known. Therefore, the inpainting algorithm regions are circles of 3mm in diameter with the ideal fiducial positions. The definition of the inpainting regions is seen as the white circles in the central image in Figure 5-17. The image before and after the inpainting procedure is shown in Figure 5-17 left and right image, respectively.

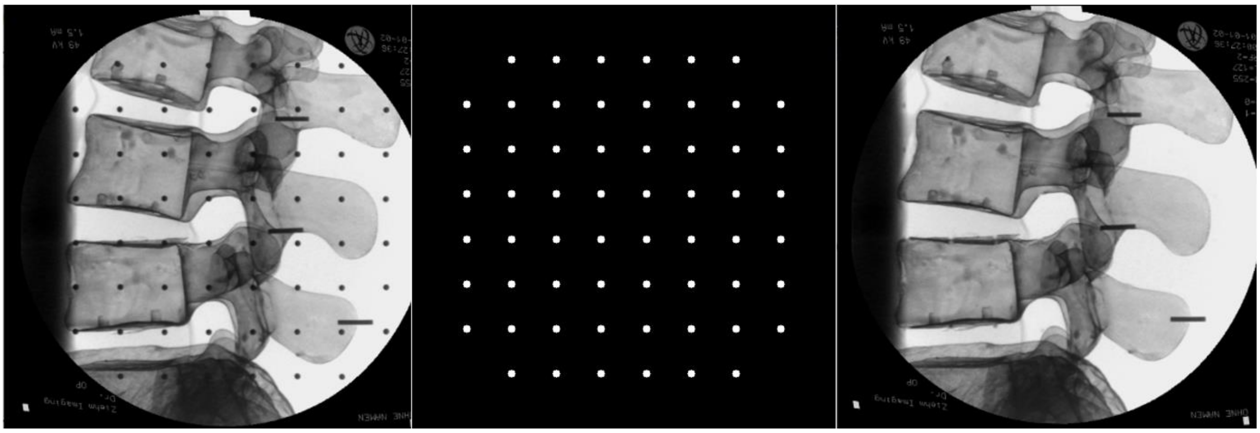

Figure 5-17. On the left side an image after the undistortion process. On the center, definition of the inpainting regions. On the right side, image after the inpainting process

Although having images without distortion helps the image similarity measurements, if the fiducials are visible after the undistortion process, the image similarity measurement will not improve. The purpose of the inpainting algorithm in the 2D/3D registration is to minimize the impact of the undistortion device fiducials in the image similarity measurement.

5.3.3.2. Finding a rigid transformation between X-ray Images with and without Distortion

Even though the undistortion device is built to be centered on the detector using some available rigid points on the C-arm, it is hardly possible to guarantee that the fiducials will be centered and aligned on the X-ray image. In other words, the center position and the alignment of the original image with respect to the image after the undistortion process is not the same. It is crucial to determine a transformation between the original X-ray image and the image after the undistortion procedure, such as the processed image has the same center and the same orientation as the original image. To illustrate the problem, an X-ray image only containing the undistortion device is taken. On the left side of Figure 5-18, the original X-ray image and the image without distortions are overlapped using a pixel-wise difference operation. It can be seen how the fiducials show a general translation and rotation.

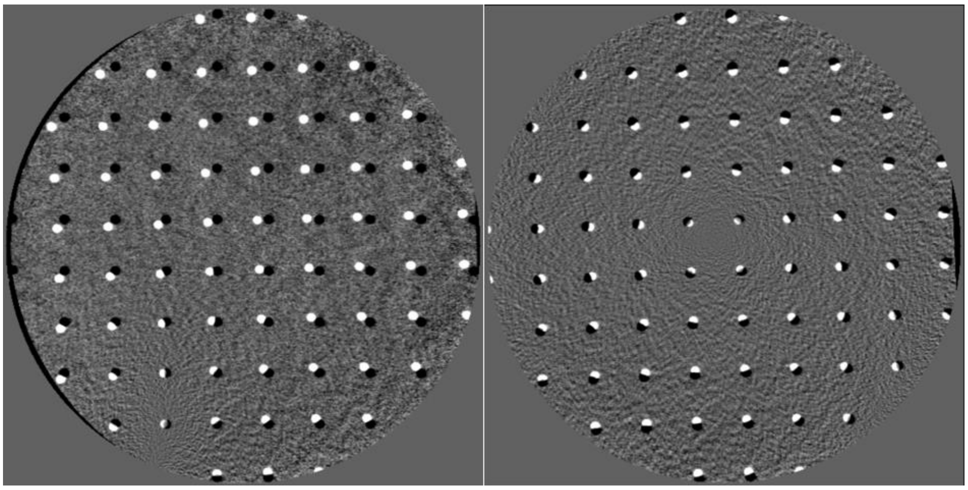

Figure 5-18. On the left side, overlapping of X-ray image before and after undistortion. On the right side, overlapping of X-ray image before and after undistortion plus translation and orientation adjustment

Distortions in X-ray images from conventional C-arms are less predominant close to the image center than in the image edges [26]. This means that the four central fiducials are, out of the whole set of fiducials, the least distorted elements. In the ideal case, i.e., image without distortion, the center point of the square described by the four central fiducials would coincide with the center of the image. In the original image, the four central fiducials defined a quadrilateral, whose center is calculated. The displacement from the image center to the quadrilateral center defines the translation from the image without distortion to the original image.

The quadrilateral corners and center in the original image and in the image after the undistortion procedure are considered. Four vectors per image are built using the quadrilateral center as the initial point and corners as the final points. The angle formed between corresponding vectors in the image with and without distortion is calculated. The rotation of the image without distortion to the original image is calculated as the average of the four angles between corresponding vectors.

Once the translation and rotation are calculated, the image without distortions is translated and then rotated using the previously calculated values. An example of the obtained result using this procedure can be seen on the right side of Figure 5-18. It can be noticed that there is an improved alignment of the original image and the image after the undistortion with respect to the image center. This additional transformation, applied to the image without distortion, is the final step to conclude the undistortion process, which was not found in the available literature as part of the undistortion. This extra step was found due to large errors in the registration accuracy, which were backtracked to a missing transformation after the undistortion process.

5.3.4. Discussion and Conclusions

The image undistortion procedure for this work is based on a mapping function created by a bi-polynomial regression of fifth order. The image undistortion rendering is made using an image warping process, which finds the pixel values using a bilinear interpolation. The undistortion device is designed to be attached to the C-arm detector. It contains 60 fiducials in an 8x8 grid shape without corners to maximize the spread of the fiducials while keeping their visibility within the X-ray. The number of fiducials used for the device is among the average number of beads in other undistortion devices in the literature.

The undistortion process results are examined using a brand-new K-wire, which can be seen as a straight metallic line. In X-rays, coming from conventional C-arms, it is possible to notice the distortion by simple observation. Five X-rays of the K-wire are taken in different positions and orientations. The X-rays are processed by the undistortion procedure, and the resulted K-wire in the images are analyzed using linear regression. All in all, it is noticed that the undistortion procedure removes the distortion effects. Based on the linear regression metrics, i.e., R-squared and the stand error of the regression, it is concluded that the K-wire fits better a line equation after the undistortion process. The standard error and the R-squared metric do not show a 100% fit, but they could be due to the edge detection process and quantization error, which is inherent to every analog to digital conversion.

An inpainting procedure is implemented to avoid adverse effects on the registration, which removes the fiducials in the images after the undistortion process.

The undistortion device is found to add a translation and rotation with respect to the original X-ray image. This rotation and translation are computed using the four central fiducials of the undistortion grid. Afterward, the rotation and translation are compensated on the image after the undistortion process.

5.4. Characterization of Reference Frames in the C-Arm

The C-arm characterization is carried out using the pinhole camera mathematical model, as seen in chapter 4.5.3. Figure 4-14 shows the standard pinhole camera, but a more illustrative pinhole camera model related to the C-arm can be seen in Figure 5-19. It can be noticed that the object to be observed lies between the optical center and the retinal plane, which does not change the mathematics developed in chapter 4.5.3.

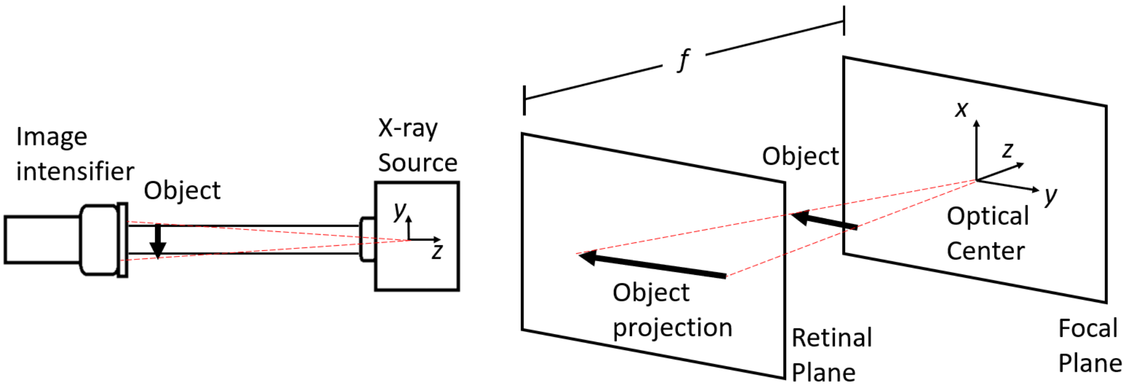

Figure 5-19. Pinhole camera model in a C-arm setting

The C-arm mathematical model expresses the intrinsic camera parameters and the X-ray source reference frame, {C-ARM}, with respect to an external frame (extrinsic camera parameters). A parametrization device is created using a set of fiducials and an RB, named CRB. The result of the parameterization binds the {C-ARM} frame with respect to {CRB}. The DLT algorithm, explained in chapter 4.5.4, is used to find the camera parameters, but some design considerations must be taken into account.

This chapter focuses on a detailed analysis, design, and implementation of a parametrization device. There are some commercial devices sold by C-arm manufacturers, e.g., Ziehm and Siemens, for recalibrating their C-arms, but a novel study analyzing every stage of the implementation of a parametrization device is developed in this section. It begins by evaluating the robustness of the implemented DLT algorithm with added Gaussian noise. The design of the parameterization device is evaluated, finding a trade-off between the number of fiducials on the device and the parametrization accuracy. The design is tested in a simulation using its 3D CAD volume and creating synthetic images using the DRR module. Then, the built device is used to examine the nature of the C-arm stiffness, as C-arms are known to undergo structural bending that causes their intrinsic parameters to vary. Finally, the quality of the manufactured parametrization device is evaluated using an innovative approach called inverse registration.

5.4.1. Evaluating the DLT Algorithm

As seen in chapter 4.5.4, the DLT algorithm solves equation (*44*) to find the camera parameters ($\boldsymbol{P}$) matrix that is decomposed into the extrinsic ($\boldsymbol{E_p}$) and intrinsic parameters ($\boldsymbol{I_p}$). A first step to test the implementation is to propose a set of N points $\mathbf{S}_4 = \{\tilde{\boldsymbol{s}}_{\mathbf{41}}, \dots, \tilde{\boldsymbol{s}}_{\mathbf{4}N} \mid \tilde{\boldsymbol{s}}_{\mathbf{4}i} \in \wp^4\}$, create a $\boldsymbol{P}$ matrix out of the proposed $\boldsymbol{E_p}$ and $\boldsymbol{I_p}$ parameters, and project the set of $\mathbf{S}_4$ points into the set $\mathbf{S}_3 = \{\tilde{\boldsymbol{s}}_{\mathbf{31}}, \dots, \tilde{\boldsymbol{s}}_{\mathbf{3}N} \mid \tilde{\boldsymbol{s}}_{\mathbf{3}i} \in \wp^3\}$. With $\mathbf{S}_4$ and $\mathbf{S}_3$, it is possible to recalculate the used matrix $\boldsymbol{P}$, call $\boldsymbol{P}'$, and decomposed it into $\boldsymbol{E_p}'$ and $\boldsymbol{I_p}'$. Testing the implementation consists of finding the difference between the proposed extrinsic and intrinsic parameters, $\boldsymbol{E_p}$ and $\boldsymbol{I_p}$, and the computed parameters, $\boldsymbol{E_p}'$ and $\boldsymbol{I_p}'$. Since the number of points in $\mathbf{S}_4$ impacts the accuracy, ten sets of points are compared, whose set sizes ranges from six to sixteen points. The sets of points are randomly generated, but constraining that no more than four points belong to the same plane to avoid singularities in the DLT algorithm [31].

The comparison is run 1000 times. In each iteration a new $\boldsymbol{P}$ matrix is generated by a random uniform distribution and used to test the algorithm described above in the ten sets of points. The generation of the $\boldsymbol{P}$ matrix is delimited by boundaries imposed on the intrinsic and extrinsic parameters. The Boundaries are defined in Table 5-2. These boundaries are selected to cover the range that the intrinsic and extrinsic parameters of a real C-arm could have.

Table 5-2. Boundaries for generating the $\boldsymbol{P}$ matrix to evaluate the DLT algorithm

Type	Parameter name	Units	Min value	Max value
Intrinsic parameters	Focal length	mm	500	1000
	Optical center X	px	-284	284
	Optical center Y	px	-284	284
Extrinsic parameters	Position X	mm	-100	100
	Position Y	mm	-100	100
	Position Z	mm	100	900
	Angle X	deg	-179	180
	Angle Y	deg	-179	180
	Angle Z	deg	-179	180

For the intrinsic parameters, the focal length is a positive value within 500mm and 1000mm, and the optical center is the frame origin inside the retinal plane. As the available C-arm has an image resolution of 568x568px, the tests are carried out using this resolution. In the pinhole camera model, the origin is located on the image center, so the optical center range is selected from [-284, 284]. The extrinsic parameters describe the external frame pose with respect to the X-ray reference frame, {C-ARM}. The X- and Y-translations are relatively small movements up-down and left-right, with respect to the optical center, see Figure 5-19. The Z-translation describes the displacement of the axis from the optical center and normal to the retinal plane. The external frame rotation is restricted to the range [-179, 180], which describes a free rotation around the three axes, but it gives boundaries to the random generation.

The error is calculated using the (magnitude) relative error as expressed in equation (*121*), and Table 5-3 shows the mean error of each parameter over 1000 samples with ten different point sets.

$$relative\_error = \frac{|v_{true} - v_{computed}|}{|v_{true}|} \tag{121}$$

Table 5-3. DLT algorithm error using 1000 $\boldsymbol{P}$ matrices and ten points set sizes

Parameter error [%]$\times 10^{-5}$	Points set size									
	6	7	8	9	10	11	12	13	14	15
Focal length	18.42	0.26	0.07	0.02	0.02	0.02	0.05	0.02	0.06	0.01
Optical center X	72.07	0.71	0.37	0.04	0.01	0.04	0.36	0.15	0.10	0.04
Optical center Y	13.10	0.04	0.03	0.20	0.22	0.16	0.21	0.12	0.00	0.14
Position X	1.91	0.03	0.01	0.00	0.00	0.00	0.02	0.01	0.00	0.01
Position Y	2.11	0.09	0.01	0.02	0.01	0.03	0.02	0.01	0.02	0.01
Position Z	23.06	0.25	0.08	0.05	0.02	0.14	0.07	0.12	0.18	0.03
Angle X	0.02	0.03	0.01	0.01	0.00	0.01	0.03	0.02	0.01	0.02
Angle Y	0.01	0.01	0.01	0.36	0.00	0.00	0.00	0.00	0.00	0.00
Angle Z	0.22	0.12	0.03	0.21	0.00	0.00	0.00	0.00	0.00	0.00
Total mean error	14.55	0.17	0.62	0.10	0.03	0.05	0.08	0.05	0.04	0.03

The performance of the DLT algorithm gives no doubt as the worst error, when using a set of six points, has an average absolute error over the computed parameters of $14.548 \times 10^{-5}\%$.

5.4.2. Evaluating the DLT Algorithm Simulating Real-Life Conditions

One aspect to consider is that in the previous experiment, the measurements in both sets, i.e., $\mathbf{S}_4$ and $\mathbf{S}_3$ are known with infinite precision. However, this is not the case when the fiducials centers must be computed from X-ray images. As an image consists of pixels, the units are discrete, but the actual center of a fiducial can be in a sub-pixel position. The information to obtain the center of a fiducial in the image is discrete, i.e., pixel intensities are only defined in discrete locations. The fiducial center detection contains an intrinsic quantization error, which is at most 0.5px but to assume additional detection errors, the typical detection error is assumed as one pixel. For this reason, a similar experiment is carried out as in section 5.4.1, but adding Gaussian noise with a standard deviation of 1px to represent the measurement error of the fiducial center detection. As in section 5.4.1, starting with a set $\mathbf{S}_4$, it will be projected using a $\boldsymbol{P}$ matrix. The result once again is the set $\mathbf{S}_3$. To this set, the gaussian noise will be added, obtaining $\mathbf{S}_3' = \{\tilde{\boldsymbol{s}}_{\mathbf{31}}', \ldots, \tilde{\boldsymbol{s}}_{\mathbf{3}N}' \mid \tilde{\boldsymbol{s}}_{\mathbf{3}i}' \in \wp^3\}$. Using $\mathbf{S}_3'$ and $\mathbf{S}_4$, a similar procedure as in section 5.4.1 is followed to obtain the $\boldsymbol{P}'$ matrix and decomposed into $\boldsymbol{E}_{\boldsymbol{p}}'$ and $\boldsymbol{I}_{\boldsymbol{p}}'$. The results of the experiment are shown in Table 5-4.

Table 5-4. DLT algorithm error using 1000 $\boldsymbol{P}$ matrices, ten points set sizes, and gaussian noise $\sigma = 1px$ in the 2D positions

Parameter error [%]	Points set size									
	6	7	8	9	10	11	12	13	14	15
Focal length	2.91	0.44	0.38	0.32	0.31	0.27	0.27	0.25	0.23	0.21
Optical center X	97.73	23.78	15.46	13.03	10.47	11.36	12.32	8.22	11.57	9.19
Optical center Y	447.20	24.16	49.21	24.91	40.94	35.40	19.43	28.26	16.65	20.90
Position X	7.27	6.21	2.77	2.43	3.00	2.66	3.13	2.72	3.34	1.83
Position Y	22.92	3.09	2.26	3.46	2.19	2.15	2.24	1.83	1.72	1.91
Position Z	3.77	0.42	0.35	0.30	0.30	0.27	0.27	0.24	0.23	0.20
Angle X	4.83	0.58	0.38	0.30	0.42	0.32	0.34	0.29	0.34	0.28
Angle Y	3.94	0.40	0.27	0.21	0.27	0.21	0.22	0.20	0.22	0.19
Angle Z	3.06	0.22	0.15	0.12	0.12	0.10	0.10	0.11	0.10	0.09
Total mean error	65.96	6.59	7.91	5.01	6.45	5.86	4.26	4.68	3.82	3.87

The first to note is that the error is more than six orders of magnitude higher when considering noise than in the ideal case. Additionally, a tendency in a lower error can be observed when the number of fiducials increases. This simulation shows that a device with more than twelve fiducials has an error lower than 5%. It can also be deduced that running simulations with bigger point set sizes could lead to better results. The literature states that an optimal result is achieved with more than 35 fiducials [40]. However, many such fiducials will make an impractical solution for the 2D/3D registration, as the number of fiducials will substantially obstruct the body segment in the X-ray image.

Besides the previous consideration, there is still another error source to contemplate before carrying on with an analysis to select the fiducial set size for developing a device. The 3D list of points $\mathbf{S}_4$ is taken from the design of the device. The location of the fiducials on the real device differs from the list of points due to the manufacturing process. This manufacturing added error is simulated by adding Gaussian noise to the 3D list of points obtaining $\boldsymbol{S}_4^{'} = \{\tilde{\boldsymbol{s}}_{\boldsymbol{41}}^{'}, \dots, \tilde{\boldsymbol{s}}_4^{'} \mid \tilde{\boldsymbol{s}}_{4i}^{'} \in \wp^4\}$. The set $\boldsymbol{S}_4^{'}$ is projected to $\wp^3$ finding $\boldsymbol{S}_3$. When manufacturing devices using numerical control machining tools, the manufacturing accuracy can be lower than 10µm. In this case, the device is planned to be fast-prototyped using a 3D printer and manually assembled. The fiducials centers are calculated with the navigation system using a pointer; therefore, the standard deviation of the Gaussian noise is set to 0.1mm as the navigation system accuracy, see section 5.1.1. Using $\boldsymbol{S}_4$ and $\boldsymbol{S}_3$, a similar procedure as before is followed to obtain the $\boldsymbol{P}^{'}$ matrix and decomposed into $\boldsymbol{E}_{\boldsymbol{p}}^{'}$ and $\boldsymbol{I}_{\boldsymbol{p}}^{'}$. The results of the experiment are shown in Table 5-5.

Table 5-5. DLT algorithm error using 1000 $\boldsymbol{P}$ matrices, ten points set sizes, and gaussian noise $\sigma = 0.1mm$ in the 3D positions

Parameter error [%]	Points set size									
	6	7	8	9	10	11	12	13	14	15
Focal length	10.41	0.59	0.40	0.40	0.39	0.36	0.53	0.45	0.41	0.49
Optical center X	175.84	60.06	36.62	41.15	38.01	33.57	41.63	48.87	38.21	49.32
Optical center Y	416.55	60.19	33.51	39.24	31.51	27.61	51.77	46.98	37.32	37.30
Position X	14.71	3.16	2.98	3.03	2.82	2.90	2.87	2.73	2.93	3.46
Position Y	66.57	5.43	7.31	8.49	7.94	7.17	6.59	7.04	7.15	4.14
Position Z	4.80	0.52	0.32	0.32	0.33	0.29	0.41	0.35	0.32	0.38
Angle X	11.11	1.74	0.96	0.94	0.93	1.00	1.25	0.95	0.96	1.25
Angle Y	14.31	1.94	1.21	0.98	1.00	1.07	1.15	1.04	1.00	1.38
Angle Z	17.51	2.13	1.47	1.02	1.07	1.13	1.04	1.13	1.05	1.51
Total mean error	81.31	15.08	9.42	10.62	9.33	8.34	11.92	12.17	9.93	11.03
Total mean error without optical center	19.92	2.21	2.17	2.09	2.07	1.99	1.98	1.96	1.97	1.80

At first glance, this experiment result shows a high sensitivity over the 3D set as the mean error is more significant than in the last experiment, almost twice in every set size. However, the most significant contributor of the mean error is the optical center, miscalculated due to the error in the 3D data set. This detail is considered in further stages when designing the solution for the physical device. This experiment shows that the DLT algorithm can compute the C-arm parameters with an error below 2% when the point set size is larger than ten. It is worth mentioning that the 2% of mean error is calculated without considering the optical center.

As a final step to select an appropriate fiducial set size, an additional experiment is run, mixing both previous phenomena. To the 3D list of points $\boldsymbol{S}_4$, Gaussian noise with standard deviation of 0.1mm is added, obtaining $\boldsymbol{S}_4^{'}$. This set is projected to $\wp^3$ finding $\boldsymbol{S}_3$. Gaussian noise of standard deviation of 1px is added to $\boldsymbol{S}_3$, obtaining $\boldsymbol{S}_3^{'}$. Using $\boldsymbol{S}_3^{'}$ and $\boldsymbol{S}_4$, a similar procedure is followed to obtain the $\boldsymbol{P}^{'}$ matrix and decompose it into $\boldsymbol{E}_{\boldsymbol{p}}^{'}$ and $\boldsymbol{I}_{\boldsymbol{p}}^{'}$. The results of the experiment are shown in Table 5-6.

Table 5-6. DLT algorithm error using 1000 $\boldsymbol{P}$ matrices, ten points set sizes, and gaussian noise $\sigma = 0.1mm$ in the 3D positions and $\sigma = 1px$ in the 2D positions

Parameter error [%]	Points set size									
	6	7	8	9	10	11	12	13	14	15
Focal length	52.04	0.90	0.66	0.61	0.55	0.57	0.50	0.53	0.51	0.50
Optical center X	715.29	149.90	68.40	92.77	77.61	61.60	49.49	56.60	51.57	49.94
Optical center Y	4901.76	175.64	37.39	58.22	49.20	54.14	46.26	45.53	55.83	58.55
Position X	91.66	4.37	4.34	5.63	4.32	4.20	4.51	4.10	3.64	4.09
Position Y	44.42	8.17	6.96	8.16	6.47	7.15	6.24	6.43	6.61	5.31
Position Z	9.64	0.68	0.58	0.55	0.48	0.47	0.45	0.47	0.45	0.42
Angle X	25.88	6.67	5.34	7.85	5.17	11.81	3.78	4.47	4.16	4.77
Angle Y	19.30	4.31	3.07	4.41	3.08	6.31	2.47	2.73	2.56	2.87
Angle Z	12.71	1.95	0.81	0.98	1.00	0.81	1.17	0.99	0.96	0.98
Total mean error	652.52	39.18	14.17	19.91	16.43	16.34	12.77	13.54	14.03	14.16
Total mean error without optical center	36.52	3.86	4.03	3.11	3.01	4.47	2.73	2.82	2.70	2.70

As a result of two added gaussian noise sources, the mean error of the solution increases compared to the previous run experiments. It can be observed that, from the point set size greater than twelve, the mean error without optical center lies below 3%.

The last experiments have shown that the error tends to decrease with a larger set size. A device with more than eight fiducials already displays an error closer to the largest set size tried, but after size twelve is more evident to see a slower improvement. These mathematical experiments display promising results with devices with more or equal than twelve fiducials. Consequently, a twelve-fiducial device is built as it represents a good trade-off between accuracy in the calculation of the C-arm parameters and space taken in the X-ray image due to the number of fiducials.

5.4.3. Designing and Simulating a Parametrization Device to Perform the C-arm Resectioning

The device is designed with four beams. Three fiducials are installed on a beam, and the positions of the fiducials are different in each beam. This keeps a different fiducial layout in every beam. Each beam is installed with a different angle out of a common point to avoid more than four fiducials on a plane. In Figure 5-20, it is possible to see the CAD of the parametrization device.

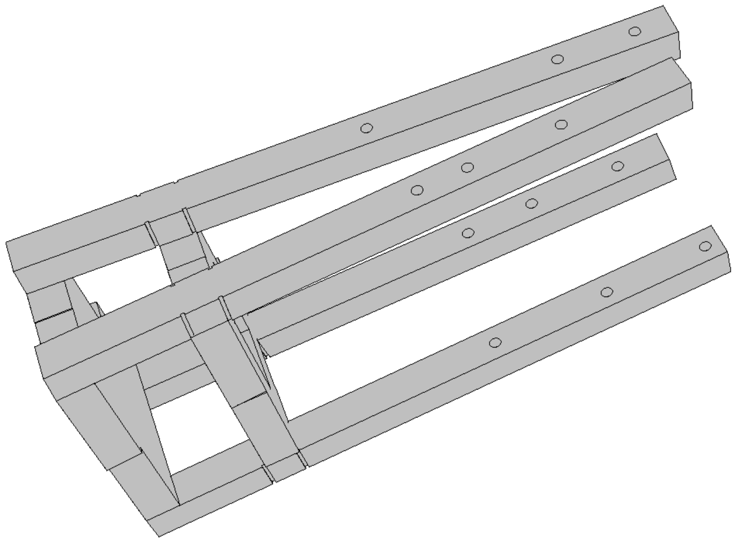

Figure 5-20. Initial design of the parametrization device

The fiducials positions are designed considering the detector diameter of the Ziehm Vario 3D C-arm, 230mm. The fiducials distribution considers that the device installation point is close to the detector. When an X-ray is taken, all of the fiducials are visible in the image. The fiducials positions in the reference frame of the 3D volume are known by construction. In context with section 5.4.1, it means the set of points $\mathbf{S}_4$ is already known.

To test the effectiveness of this design, the DRR module will be used to replace the C-arm. Using the DRR offers the additional advantage of applying the DLT algorithm in a controlled environment that uses the same operating principle as the C-arm. Since the DLT algorithm only requires the fiducials centers, the shape of the calibration device is not critical in this stage, only the distribution of fiducial in space. The input of the DRR module is fed with a DICOM file containing the fiducials distribution of the calibration device. A render of the DICOM can be seen in Figure 5-21.

Figure 5-21. Fiducial dispersion on the parametrization device

The best advantage of using the DRR at this stage is its configurability, i.e., the focal length, image resolution, and pixel spacing can be set. Those are part of the intrinsic parameters of the pinhole camera model. The DRR optical center is centered in (0,0) by definition, so it cannot be modified, but it is known. Additionally, the DICOM pose with respect to the DRR is part of the parameters that can be modified in the DRR. This previous pose can be used to build the extrinsic parameter matrix.

With the list of detected fiducials in the DRR image, i.e., $\mathbf{S}_3$ and the list of 3D points, i.e., $\mathbf{S}_4$, the intrinsic and extrinsic pinhole camera parameters can be calculated. Through another path, the DRR settings can be used to find the intrinsic and extrinsic parameters too. Both results are used to calculate the error and validate the robustness of the mechanical design of the parameterization device. The fiducial center detection is carried out by finding the mass center of the fiducial, reported in the literature as an accurate procedure for calculating the circle center [36] [164]. In this experiment, the focal length of the DRR is set to 1000mm, a common source to detector distance in C-arms. Manually in the DRR module, the output images are analyzed to determine the range where the fiducials are visible. This range can be seen in Table 5-7, which corresponds to the {DRR} reference frame shown in Figure 5-4.

Table 5-7. Boundaries for generating the $\boldsymbol{P}$ matrix to evaluate the DLT algorithm

Parameter name	Units	Min value	Max value
Position X	mm	-10	5
Position Y	mm	350	500
Position Z	mm	-10	10
Angle X	deg	-5	5
Angle Y	deg	-8	-1
Angle Z	deg	-5	5

From this range, a set of 1000 random DRR images are created using a uniform distribution. Then the DLT algorithm was run. The results are tabulated in Table 5-8. The absolute error is used, i.e., $|v_{true} - v_{computed}|$, which has the same dimension as the parameter it describes. Therefore, the set of errors of all parameters is no longer averaged, but analyzed individually. In this case, the relative error is avoided as the actual value of the optical center is zero, so the relative error becomes meaningless [165]. The angles and positions X and Z are in a range close to zero. It means an error of 1mm in a position with an actual value of 1mm gives a relative error of 100%. The relative error would lead to misinterpretations of the results for those ranges close to zero, so the absolute error can be more meaningful here.

Table 5-8. Boundaries for generating the $\boldsymbol{P}$ matrix to evaluate the DLT algorithm

Parameter name	Absolute error
Focal length [mm]	6.82
Optical center X [px]	12.31
Optical center Y [px]	11.33
External frame position X [mm]	0.86
External frame position Y [mm]	6.48
External frame position Z [mm]	0.21
External frame rotation X [°]	0.23
External frame rotation Y [°]	0.01
External frame rotation Z [°]	0.30

The results show similar behavior as in the pure mathematical simulation. The optical center is the measurement with the highest error, followed by the focal length. As the focal length has a large and constant actual value, the relative error can be computed as 0.682%. This error is higher than the focal length errors computed in the last sections, which can be attributed to two causes. One is the accuracy of fiducial center detection. The second reason, the opacity of the fiducials in the DRR images is not strong enough due to the DICOM generation process, see Appendix A.8. On the other hand, in the real device, the fiducials are made of steel, a material with high opacity under the X-rays.

The computation of the parameters with the DLT algorithm using the DRR module is considered to be within acceptable accuracy compared to the simulations performed in section 5.4.2. It has also been seen in the literature that the error in parameter estimation using a calibration device is more relevant in the focal length and the optical center [166].

One interesting feature of the estimation problem can be seen by plotting the focal length histogram, which shows that the pinhole camera parameter calculation follows a gaussian distribution. See Figure 5-22. This leads to the assumption that the mean of parameters calculated from many samples approaches the values of the actual parameters as stated by the central limit theorem [167].

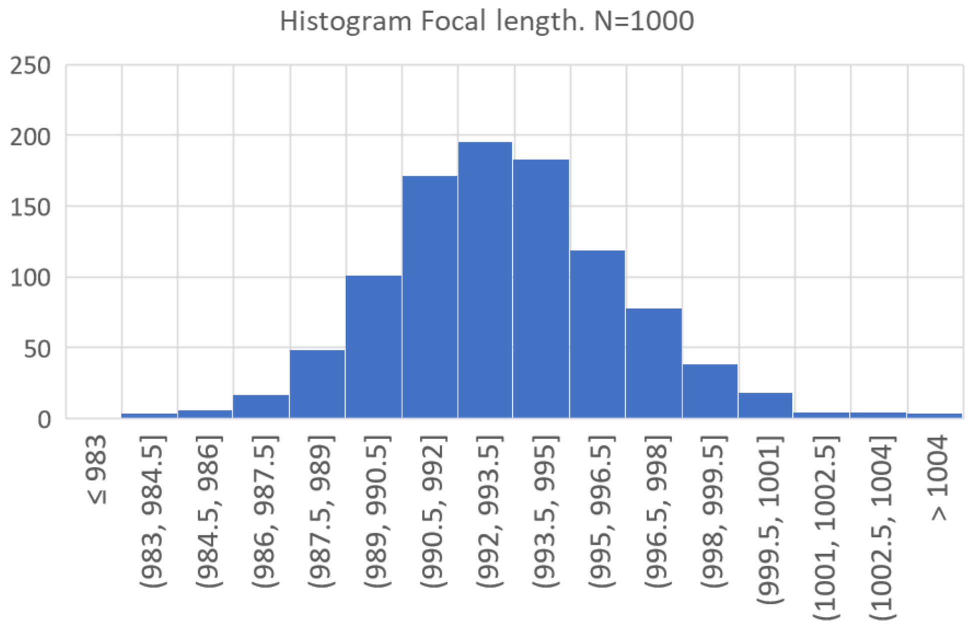

Figure 5-22. Focal length histogram using 1000 DRR images

Design of the Parametrization Device

Two considerations are followed for the design of the parametrization device. First, the device can be fixed to the C-arm or carried by an operator. Secondly, the fiducials center references are kept by an RB attached to the device, named CRB, whose placement is considered in the design stage. The entire design can be seen on the left side of Figure 5-23. The implemented device already attached to the C-arm detector can be seen in Figure 5-23, right side.

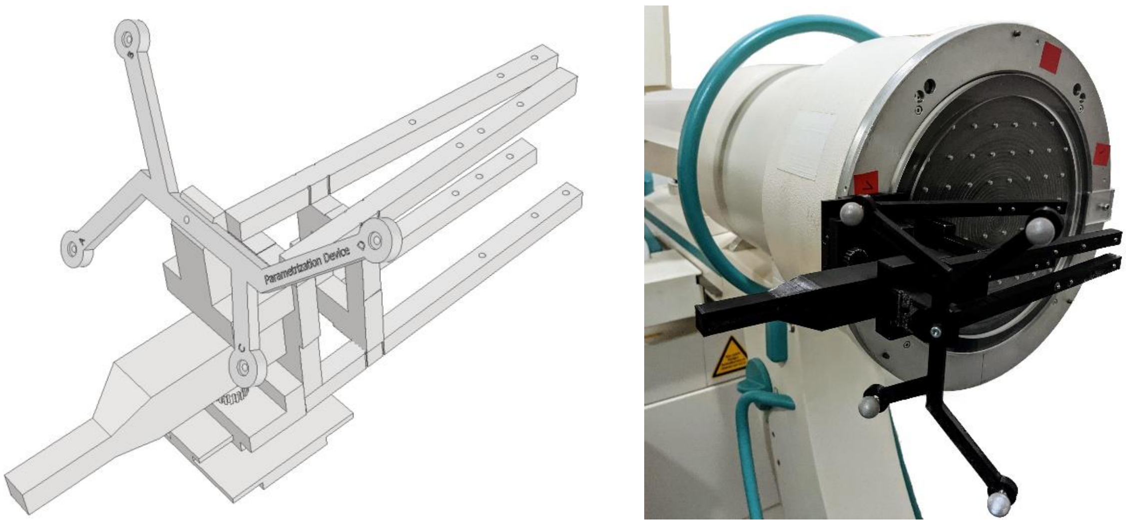

Figure 5-23. Full design of the parametrization device on the left, installed device on the right

As reported in the literature, image intensifiers C-arms suffer from distortion due to the earth's magnetic field [28]. Additionally, the C-shape structures are prone to bending, which creates fluctuations in the intrinsic parameters depending on the C-arms poses [27]. The undistortion procedure is explained in section 5.3, so only images without distortion are considered for the C-arm parametrization. The fluctuations of the C-arm intrinsic parameters are determined for a specific C-arm. The parametrization device is rigidly attached to the C-arm detector, so the intrinsic and extrinsic parameters found in consecutive images are expected to be similar.

The Ziehm Vario 3D C-arm offers the possibility to make scans. A scan takes 130 X-ray images along the C-arm arc, which has a 135° stroke. Ten of these scans are performed, varying the rotation point of the C-arm; thus creating 1300 unique images. For each of the images, the intrinsic and extrinsic parameters are calculated. One of the simplest parameters to analyze is the focal length, since the C-arm bending is expressed as a one-dimensional parameter. The parameters of the ten scans are sorted by the C-arm arc angle. The scans start at 90° from the LAT rotation, thorough AP (0°), and end at -35°. The values are further gathered in groups of 10° range. For each range, there are 100 samples, which are averaged. The plot of focal length vs. angle of the C-arm can be seen in Figure 5-24.

It can be noticed how the bending of the C-arm structure affects its focal length. From the values at 90° (LAT projection) and 0° (AP projection), the focal length is computed to be 967.17 mm and 966.99mm, respectively. The intermediate positions are shifted considerably, and the range of the angles from 0° to -35° shows a decrease in the focal length.

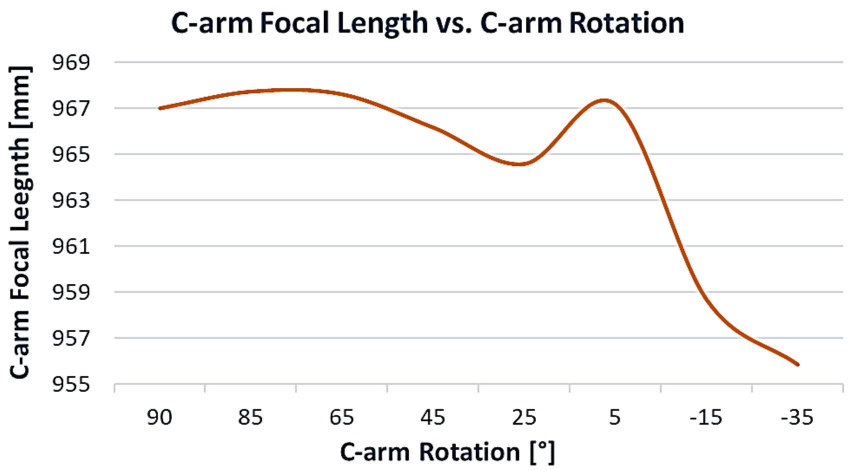

Figure 5-24. Focal length vs angle of the C-arm

5.4.4. Testing accuracy of the C-arm parametrization using the Inverse Registration Approach

For testing the accuracy of the found C-arm parameters, the following experiment is conducted:

Let a new device be introduced, see Figure 5-25, which has the following characteristics:

- There is a CT-scan from the device as a DICOM image.
- The device has seven landmarks that can be detected both physically and in its available modalities, i.e., X-ray and CT-scan.
- The device has an RB rigidly attached, which keeps its registration, i.e., ${}^{RB}\boldsymbol{T}_{DICOM}$.
- This device registration has an error smaller than 1mm. Therefore, it will be referred from now on as the well-calibrated device.

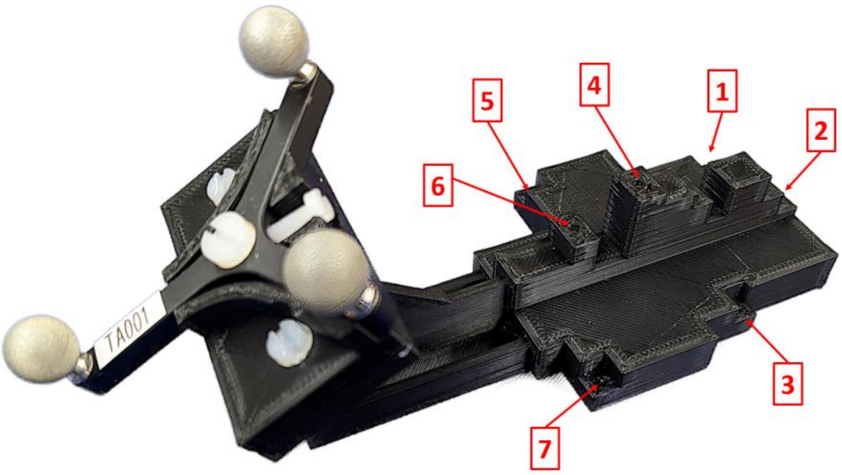

Figure 5-25. Well-Calibrated device to test C-arm calibration

Five AP and five LAT images of this device are taken using the C-arm, while taken measurements with the optical tracking system. As ${}^{RB}\boldsymbol{T}_{DICOM}$ of this device is known, ${}^{DRR}\boldsymbol{T}_{DICOM}$ can be solved from equation (*87*), obtaining the inverse registration equation:

$$ {}^{DRR}\boldsymbol{T}_{DICOM} = {}^{DRR}\boldsymbol{T}_{C-ARM} \cdot {}^{C-ARM}\boldsymbol{T}_{CRB} \cdot {}^{CRB}\boldsymbol{T}_{OTS} \cdot {}^{OTS}\boldsymbol{T}_{ARB} \cdot {}^{RB}\boldsymbol{T}_{DICOM} \qquad (122) $$

${}^{DRR}\boldsymbol{T}_{DICOM}$ is the pose that is used in the DRR module. With the DICOM of the well-calibrated device, a DRR image can be created. All the elements of equation (*122*) are known, and with an ideal ${}^{C-ARM}\boldsymbol{T}_{CRB}$, the DRR image will look identical to the X-ray. Then the quality of the found ${}^{C-ARM}\boldsymbol{T}_{CRB}$ can be inferred from the match between the DRR image and the X-ray. The evaluation of the found ${}^{C-ARM}\boldsymbol{T}_{CRB}$ is carried out using the seven available landmarks of the device. The coordinate of each landmark center is detected on the DRR and X-ray image. The Euclidian distance of each pair is used, meaning seven Euclidian distances will be calculated per test. Since ten images are taken for this experiment, 70 pairs of corresponding landmark coordinates are used to find the quality of the C-arm parametrization. The testing procedure using one of the AP images is illustrated in Figure 5-26.

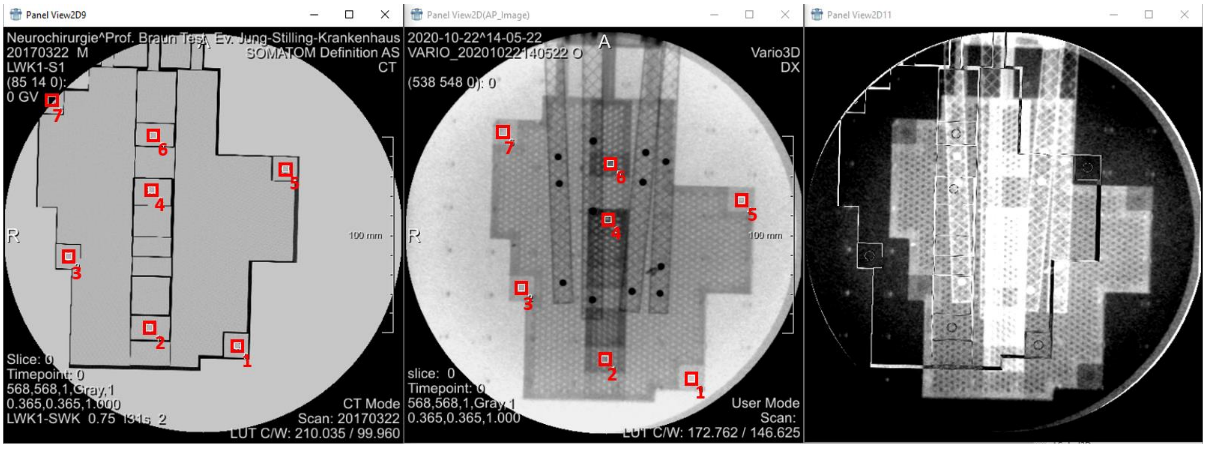

Figure 5-26. Testing C-arm parameters. On the left image, the DRR image with the detected landmarks. On the center, the X-ray image with the detected landmarks. A subtraction operation of the DRR image and the X-ray image is visible on the right

The landmarks labeling on the X-ray image can be seen in the central frame of Figure 5-26; the DRR image with the labeled landmarks can be seen on the left. On the right frame, a subtraction operation of the DRR and the X-ray image is performed to visualize the C-arm parametrization accuracy. Visually, the overlapping does not show a good match, corroborated in the results in Table 5-9.

Table 5-9. Distance landmark in X-ray image to DRR image

	Distance landmark pairs in X-ray to DRR [px]						
	Landmark						
Projection	1	2	3	4	5	6	7
AP1	30.5	30.6	30.4	30.4	30.4	30.0	30.8
AP2	33.9	33.3	33.7	33.0	34.1	33.4	33.5
AP3	38.0	37.1	38.2	37.8	37.7	38.7	38.8
AP4	40.4	40.4	40.1	39.4	40.2	40.4	40.6
AP5	33.7	34.4	34.1	34.3	34.0	34.3	34.9
LAT1	208.1	192.5	78.5	80.2	79.5	79.1	78.5
LAT2	39.9	38.9	42.4	39.2	39.6	39.2	38.3
LAT3	17.2	18.3	21.1	18.0	16.9	16.8	16.4
LAT4	5.7	5.4	5.9	5.3	4.3	3.9	4.7
LAT5	4.3	3.8	4.1	4.0	3.6	3.9	3.9
					Distance average:		35.7

It can be noticed that the distance average between corresponding landmarks is about 35.7 pixels. This average indicates a mismatching of the X-ray and the DRR image due to the C-arm parametrization. As the landmarks are distributed within the entire 3D volume of the well-calibrated device, the mismatch is a combination of rotational and translational miscalculation of the C-arm parameters. Since the distance between pairs within the same projection is similar, a parametrization error in the translation can be assumed. For example, the distance between landmarks is around 30.5px in the projection AP1. That means the error mostly comes from an inaccurate estimation of the optical center or the external frame position. Another clue comes from the images LAT4 and LAT5, where the average distance is 4.5px. To have a broader view of this phenomenon, the calculated optical center of each X-ray image is presented in Table 5-10.

Table 5-10. Optical center of the projections to determine C-arm parametrization accuracy

		AP1	AP2	AP3	AP4	AP5	LAT1	LAT2	LAT3	LAT4	LAT5
Optical center	X	71.7	75.6	84.9	87.8	85.0	165.6	96.0	37.5	-8.8	10.9
	Y	38.5	48.9	56.2	59.8	33.2	-155.0	-61.2	-38.2	-15.5	-11.0

A strong correlation between the distance of the landmarks and the optical center can be noticed. For instance, the optical center of the image LAT1 is (165.6, -155.0), the farthest from (0,0), and the average distance of the landmarks is 113.8, the largest of the entire set. The images LAT4 and LAT5 show oppositely the smaller distance between corresponding landmarks and closer optical center to (0,0).

With this observation, the experiment is elaborated further. The averaged focal length values of the AP and LAT projections, taken from Figure 5-24 angles 0 and -90°, respectively, and an optical center in the origin are used to optimize the 3D list of points during the calculation of the C-arm parameters [32]. The implemented optimizer is a simplex hill, and the delta steps are changes of 0.02mm in the 3D list of points. After optimizing the C-arm parameters, the experiment is repeated. The graphical results for the image AP1 can be seen in Figure 5-27.

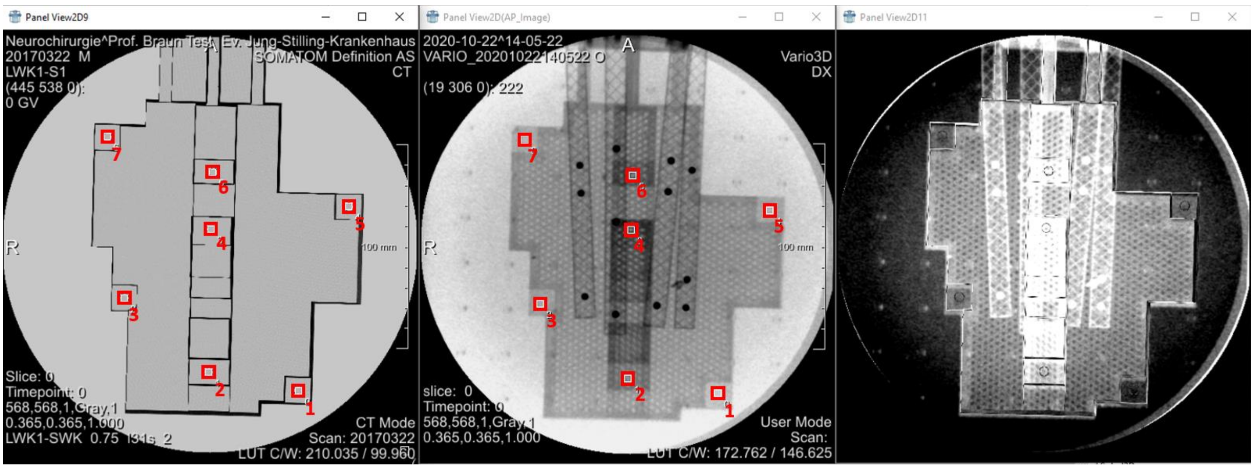

Figure 5-27. Testing C-arm optimized parameters. On the left image, the DRR image with the detected landmarks. On the center, the X-ray image with the detected landmarks. A subtraction operation of the DRR image and the X-ray image is visible on the right

In this case, the results look more accurate in comparison with Figure 5-26, and numerically they can be seen in Table 5-11.

Table 5-11. Distance points in X-ray image to DRR image

	Distance points in X-ray to DRR [px]						
	Landmark						
Projection	1	2	3	4	5	6	7
AP1	1.8	0.9	2.5	0.8	1.8	0.8	2.3
AP2	1.9	1.2	2.6	1.3	1.7	2.1	2.5
AP3	1.6	1.0	2.2	0.6	2.4	1.8	2.0
AP4	2.3	2.6	2.6	0.6	1.8	2.3	2.2
AP5	2.1	2.6	1.9	0.7	1.3	1.9	2.2
LAT1	2.2	3.0	3.4	4.0	3.2	3.8	5.2
LAT2	2.9	3.9	3.0	3.8	2.7	3.3	4.9
LAT3	3.5	2.7	3.6	2.4	3.2	3.8	3.5
LAT4	2.7	2.8	2.0	1.9	3.1	3.1	3.0
LAT5	1.7	2.7	2.1	2.2	3.0	2.9	3.7
						Distance average:	2.5

After optimizing the camera parameters, the distance average is 2.5px compared with 35.7px when the parameters are calculated without additional optimization. The results in Table 5-11 indicate that the current approach can find a C-arm camera matrix, which projects a 3D point in 2D, matching the actual device with a difference of 2.5px on average. As the pixel spacing of the Ziehm Vario 3D is 0.345mm/px, the average distance error is 0.8625mm. This result is considered suitable for the C-arm parametrization. Therefore, every X-ray image used in the registration process uses this optimization procedure on the 3D list of points to find the C-arm parameters. That means each X-ray image calculates a ${}^{C-ARM}\boldsymbol{T}_{CRB}$, which results from running the DLT algorithm plus the final optimization stage.

5.4.5. Discussion and Conclusions

As the C-arm images are formed from an ideal focal X-ray source, the C-arm can be mathematically modeled using the pinhole camera model. With this approach, the intrinsic and extrinsic parameters of the C-arm can be known. The intrinsic parameters refer to the constrained parameters, which ideally are unique for each device, but that is found to be pose-dependent in practice. The extrinsic parameters represent the transformation from the {C-arm} frame to the external frame describing the positions of the fiducials {CRB}.

To find the camera parameters, the DLT algorithm is evaluated mathematically and in simulation. Using the DRR module, it is found that the DLT algorithm gives a parametrization relative error of about 0.68% when detecting the focal length.

The accuracy of the results also depends on the number of fiducials that form the parametrization device. However, a trade-off between the number of used fiducials and accuracy must be found since the more fiducials there are, the more the body segment is obstructed. After some analysis, it is found that twelve fiducials give enough accuracy while keeping the image obstruction to a minimum extend.

The algorithm to detect the center of the fiducials is selected based on a literature review that states the advantage of using a mass center approach instead of a geometrical center computation. The implemented detection algorithm is analyzed, and it shows that pinhole camera parameter calculation can be described with a Gaussian distribution.

A set of scans is executed, covering the entire motion range of the C-arm. The results of the scans are used to plot the curve focal length vs. C-arm angle, showing a volatile focal length over the motion range. Presumably, the C-arm bending is the cause of the focal length variation.

A test for the C-arm parametrization is carried out using a well-calibrated device with known landmarks and a novel inverse registration approach. The test result provides a pose for the DRR module, which creates a DRR image from the DICOM of the well-calibrated device. Ideal C-arm parameters make a perfect match between the DRR and the X-ray image. The error in estimating the parameters is measured based on the Euclidian distance between corresponding landmarks in the DRR and X-ray image. The initial results give an average distance among corresponding landmarks of 35.7px. Later, an optimizer is implemented to improve the C-arm parametrization. In this final stage, the average distance of corresponded landmarks between DRR images created with the found camera parameters and the actual X-ray images reduces from 35.7px to 2.5px. This disparity between imaging modalities is considered suitable for the C-arm parametrization.

5.5. Evaluation and Selection of Image Similarity Measurement (Merit) Functions and Optimization Algorithms

In section 4.7.2 and 4.7.3, twelve image similarity measurements and five optimizers are introduced, respectively. An image similarity measure is used in tandem with an optimizer, following the procedure explained in sections 4.7.4 and 5.2 to find the pose that minimizes the 2D/3D registration cost function. The nearness of the initial pose to the actual registration pose impacts the 2D/3D registration accuracy [168], and it is influenced by the body segment to be registered, the used image measurement, and the optimization algorithm [124] [6] [116].

This section discusses the unique outcome of selecting the image similarity measurement and the optimization algorithm for the specific lumbar spine intensity-based 2D/3D registration using pre-operative CT-scan imaging and intraoperative X-ray images [169]. Seven combinations of optimizers connected in cascade are included in the evaluation, attempting to find a novel optimizer for the registration problem. The selection process is based on an experimental procedure that combines AP and LAT images to execute several 2D/3D registrations. With the help of some predefined landmarks on the phantom, the registration accuracy is calculated using the procedure in section 5.2.7 and sorted, such as the results show the performance of specific similarity measurements and optimizers.

5.5.1. Experiment Set up for the Evaluation and Selection of Image Similarity Measurement Functions and Optimization Algorithms

The development of the 2D/3D registration is explained in detail in section 5.2, and it is mentioned that the optimal setup uses two images from perpendicular projections [70]. Consequently, the 2D/3D registration achieves optimal results if carried out using in tandem one AP and one LAT image.

The implemented intensity-based registration using an iterative method requires an optimizer and an image similarity measurement. In section 4.7.2, twelve similarity measurements are described, and five optimizers are introduced in section 4.7.3. Based on some preliminary tests, the BN optimizer shows the most promising results. Based on that, some additional optimization algorithms are implemented as a cascade of two (two-stage) optimizers, using the BN optimizer in the final stage. That means, after carrying out a registration, an additional registration is automatically executed using the result of the first registration as the initial pose for the second optimizer. The following two-stage optimization algorithms are implemented: GRDE + BN, AD + BN, AG + BN, and Adams + BN. Also, three-stage optimization algorithms are implemented, finishing in GRDE + BN. These combinations are AD + GRDE + BN, AG + GRDE + BN, and Adams + GRDE + BN. All in all, twelve optimizers are implemented and tested.

A spine phantom containing the sacrum bone and the vertebra L1 to L5 is the test object. A CT-scan of the phantom is also available for executing the registration. The phantom and its CT-scan rendering can be seen in Figure 5-28. These phantoms can be found commercially, and they have similar X-ray absorption as real bones, so their X-ray images are comparable to X-ray images of real bones.

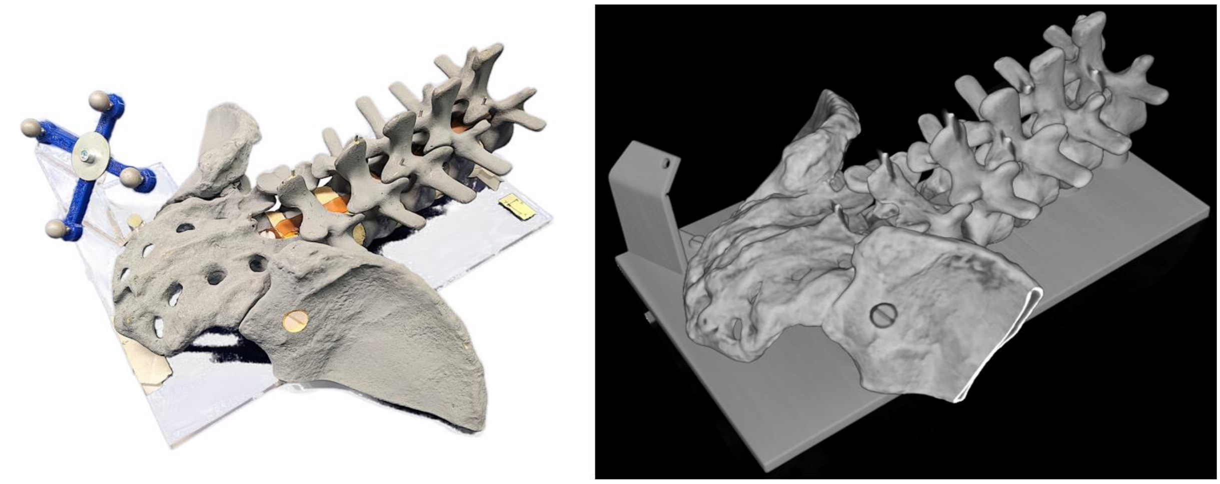

Figure 5-28. Phantom for selecting the image similarity and optimizer. Actual phantom on the left, CT-scan render on the right

5.5.2. Experiment Execution

Five AP and five LAT X-ray images of the phantom are taken using the C-arm. Since one AP and one LAT image are required for the registration process, a total combination of 25 AP and LAT images is created. For each of these 25 combinations, the registration process is executed using the twelve different optimizers and the twelve different similarity measurements. That means 144 registrations are carried out for each of the 25 combinations of the AP and LAT images. All in all, 3600 registrations are performed for this test.

Using a DICOM viewer, it is possible to extract the 3D coordinates of some landmarks in the DICOM volume, i.e., ${}^{DICOM}\boldsymbol{T}_{Landmark}$. The selection process can be seen in Figure 5-29.

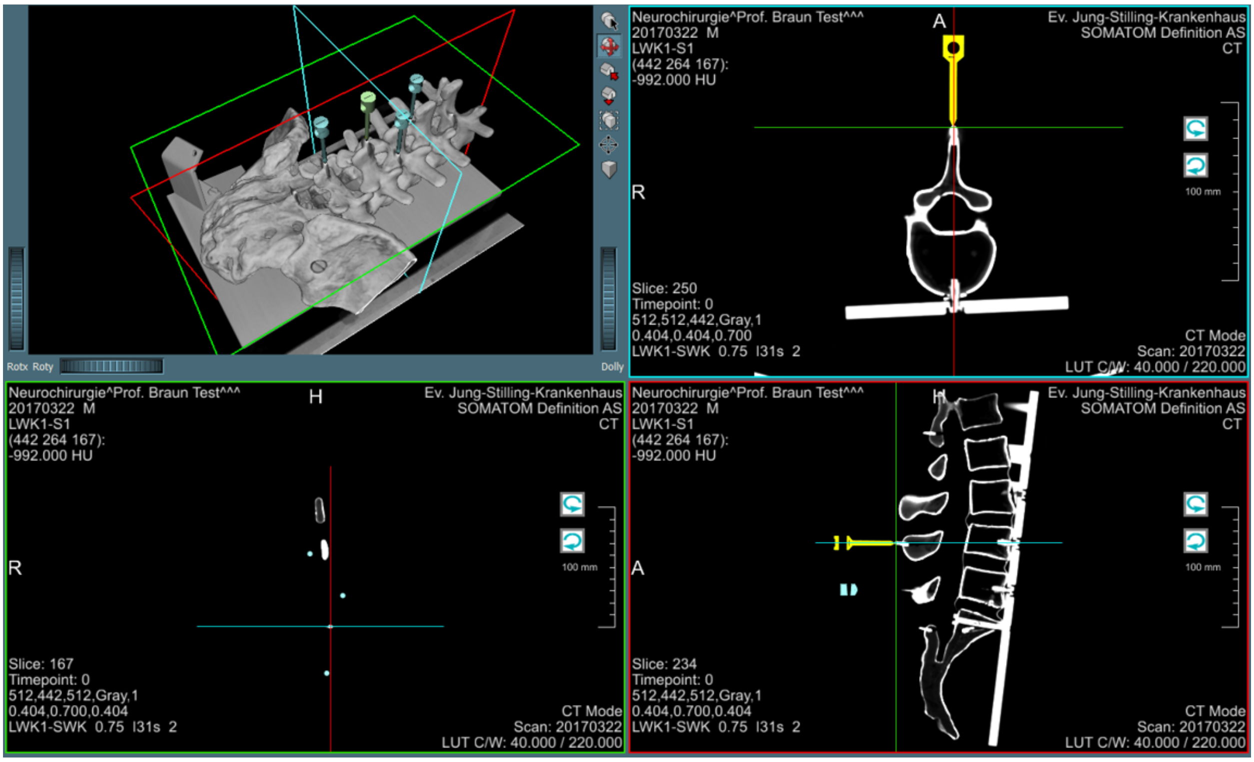

Figure 5-29. Setup to find ${}^{DICOM}\boldsymbol{T}_{Landmark}$

Likewise, the same landmark can be pinpointed on the phantom with the help of the navigation system, a pointer, and an RB attached to the phantom, i.e., ${}^{ARB}\boldsymbol{T}_{PRB}$. This setup can be seen in Figure 5-30. When the pointer is touching the landmark, the position of {PRB} is considered as {landmark}, finding ${}^{ARB}\boldsymbol{T}_{Landmark}$. The selected landmarks are steel posts installed on the phantom that can be distinguished easily.

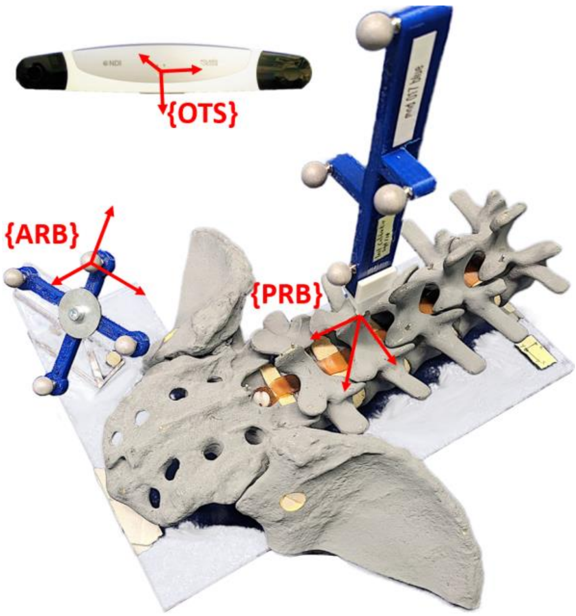

Figure 5-30. Setup to find ${}^{ARB}\boldsymbol{T}_{Landmark}$, equals to ${}^{ARB}\boldsymbol{T}_{PRB}$

Each of the 3600 registrations is evaluated using equation (*104*) from section 5.2.7. It is worth mentioning that each registration is assessed using four landmarks, which are averaged to find each registration accuracy. Depending on the similarity measurement, an iteration takes from 60ms up to 180ms. The number of iterations depends on the selected optimizer and the initial pose. During these tests, BN required an average of 12 iterations per registration, GD four iterations, and AD five iterations. That means one registration can take as short as 240ms but long as 2160ms. The computation times are not considered for the evaluation as the primary objective of the registration is to deliver accurate results, and the most critical time is acceptable. Two decision tables, which are essential for any decision-making method, are built with the registration results [170]. The decision tables are created based on the desired component to be determined, i.e., optimizer and image similarity measurement. The option that generates the smallest RMS error is selected as the most suitable component.

5.5.3. Finding a Suitable Optimization Algorithm for the Intensity-Based 2D/3D Registration for Lumbar Spine Surgery

For this selection, the decision table, Table 5-12, is built with the optimization algorithms as the rows and the RMS error of each landmark as columns. Each landmark error is created from 300 measurements, i.e., registration on the combination of five AP and five LAT images using 12 Image similarity measurements. The final average in the last column is the average over the four RMS landmark errors.

Table 5-12. Decision Table for selecting optimizers

Optimizer	RMS error Landmark1 [mm]	RMS error Landmark2 [mm]	RMS error Landmark3 [mm]	RMS error Landmark4 [mm]	Averaged RMS error [mm]
AD	3.39	3.48	2.41	2.32	2.90
AD + BN	2.99	3.05	2.07	2.11	2.55
AD + GRDE + BN	2.98	3.04	2.06	2.10	2.55
AG	3.68	3.74	2.63	2.48	3.13
AG + BN	3.02	3.07	2.07	2.15	2.58
AG + GRDE + BN	2.99	3.04	2.06	2.15	2.56
Adams	5.18	5.28	4.46	4.96	4.97
Adams + BN	3.65	3.71	2.76	2.92	3.26
Adams + GRDE + BN	3.73	3.81	2.87	3.05	3.36
BN	2.86	2.84	1.83	2.01	2.38
GD	4.84	4.87	3.71	3.44	4.22
GD+ BN	3.23	3.31	2.33	2.46	2.83

Table 5-12 shows that BN gives the best average RMS error, 2.38mm, followed by AD+BN and AD+GRDE+BN with 2.55mm, and by AG+GRDE+BN and AG+BN with 2.56mm and 2.58mm, respectively. It is concluded that the best optimizer for this work is BN. BN is not the most popular choice in literature for 2D/3D registration because it is computationally inefficient, i.e., it evaluates the cost function more than any other optimization algorithm before it reaches its minimum. On the other hand, one of the most critical elements to consider in a minimally invasive surgery is accuracy. In this case, the most accurate result is achieved through the highest computational-demanding method, which takes some hundreds of milliseconds more.

5.5.4. Finding Suitable Image Similarity Measurement for the Intensity-Based 2D/3D Registration for Lumbar Spine Surgery

The decision table is built in this case with the similarity measurements as the rows, while the columns have the RMS error of each landmark. See Table 5-13. The error of each landmark is not calculated over all the optimizers; instead, the best five optimization algorithms found in section 5.5.3 are used. While using the best five optimizers, there are still 125 samples per similarity measurement. Since the decision table is made with the optimizers that give the smallest landmark error, the selected image similarity measurement is assumed to lead to the lowest landmark RMS error.

Table 5-13. Decision Table for selecting Similarity Measurements

Similarity Measurement	RMS error Landmark1 [mm]	RMS error Landmark2 [mm]	RMS error Landmark3 [mm]	RMS error Landmark4 [mm]	Averaged RMS error [mm]
CR	2.80	2.86	1.93	2.12	2.43
GC	2.66	2.82	1.90	2.04	2.36
GD	3.12	3.08	2.15	2.26	2.65
MI	2.66	2.74	1.80	1.97	2.29
NACC	3.32	3.25	2.21	2.24	2.75
NCC	3.21	3.22	2.24	2.21	2.72
NMI	2.84	2.83	1.92	2.03	2.41
PI	2.76	2.80	1.93	2.04	2.38
SAD	3.25	3.26	2.11	2.17	2.70
SLNC	2.86	2.98	1.85	1.99	2.42
SSD	3.35	3.39	2.28	2.13	2.79
VWSLNC	2.80	2.85	1.90	2.07	2.40

It can be seen from Table 5-13 that MI is the similarity measurement giving the most accurate registrations. MI gives 2.29mm of averaged RMS error, followed by GC with 2.36mm. The obtained results agree with the literature, which points out the effectivity of these similarity measurements for 2D/3D registration applications [123] [171] [172].

5.5.5. Discussion and Conclusions

The registration error is found using a spine phantom, which has distinguishable landmarks in both the physical device and its CT, i.e., the landmark coordinate can be found. The error of the registration due to the translation mismatch is found using the procedure in section 5.2.7.

The evaluation is carried out by comparing the twelve similarity measurements and twelve optimization algorithms described in sections 4.7.2 and 4.7.3. The evaluation is created using 25 registration, made from combining five AP and five LAT images. In total, 3600 registrations are executed using the combination of optimization algorithms and image similarity measurements in each pair of AP and LAT images.

Best Neighbors is found as the image similarity measurement that gives the most accurate results in our application. BN is not commonly used in literature as it is one order of magnitude more expensive computationally than other optimizers. However, the increase from some hundreds of milliseconds to some seconds is not as critical as decreasing the registration accuracy.

Mutual information and gradient correlation give the most accurate registration for the specific lumbar spine intensity-based 2D/3D registration using pre-operative CT-scan imaging and intraoperative X-ray images. The obtained similarity measurement and optimization algorithm are considered a novel result due to the specific application and implementation discussed in this work [169]. It is worth noting that the image similarity and optimization algorithm selection are aligned with the 2D/3D registration literature when evaluating image similarity measurement in other body segments [123] [171] [172].

5.6. DRR Module Development Using Parallel Computing

The DRR module is one of the most fundamental components of the intensity-based 2D/3D registration. It is used to project the 3D CT volume into a 2D image based on a ray-casting approach. See section 3.5. Some modifications have been done to the ray-casting algorithm to make it suitable for radiological applications, known as the Siddon-Jacobs algorithm [104].

The DRR algorithm must generate images as fast as possible since the intensity-based 2D/3D registration approach requires twelve image measurements per iteration; consequently, it generates twelve DRR images per iteration.

Some implementations of the Siddon-Jacobs algorithm have been done in different frameworks in the past years [173] [174]. Although an own implementation is carried out in this work, a DRR module from a trustworthy source is used as ground-truth for validating the resulting images and testing the rendering times [174].

This section evaluates the performance of the ground-truth DRR module, creating the requirements for a new DRR module, which must maintain the same image quality with faster rendering times. With these requirements, the 2D/3D registration would keep the same accuracy with reduced time. As execution time vs. quality is usually mutually exclusive, a fast-parallel computing approach is developed using a novel pixel-step method [175]. The rendering times of the ground-truth DRR and the newly implemented module with a parallel computing approach are compared. The performance values indicate that rendering time decreases by more than three orders of magnitude when using the parallel approach.

5.6.1. Native and Serial DRR Module

The programming language used during the development of this work is C++. A medical framework developed by Fraunhofer called MeVisLab includes a ready-to-use DRR module [174]. The rendered images and the rendering times of this module are used as the ground-truth metrics of the DRR module.

A CT-scan of a phantom is used in the 2D/3D registration process, as Figure 5-28 shows. The same CT volume (in DICOM format) is used as the 3D object input for the DRR module. The DICOM file has a size of 512x512x442 voxels, which expresses its dimension in X-, Y- and Z-Axis, respectively.

The theory behind the DRR module is explained in section 4.7.1. Creating a DRR image requires three translations and three rotations gathered in $\vec{\boldsymbol{p}} \in \mathbb{R}^6$ as defined in *(32)*. The vector $\vec{\boldsymbol{p}}$ expresses the transformation from {DRR} to {DICOM}, also represented as ${}^{DRR}\boldsymbol{T}_{DICOM}$. The ray-casting algorithm goes through the volume and computes the line integral using the voxels in its path. Consequently, the computation time depends on the number of voxels to evaluate, which are a function of the CT-scan size and the ray trajectory that depends on $\vec{\boldsymbol{p}}$. Figure 5-31 depicts the difference in the ray-path length inside the volume depending on $\vec{\boldsymbol{p}}$. $\vec{\boldsymbol{p}}_{\boldsymbol{1}}$ depicts a shorter path than $\vec{\boldsymbol{p}}_{\boldsymbol{2}}$, which translates into a shorter rendering time for $\overrightarrow{\boldsymbol{p}_{\boldsymbol{1}}}$ than $\vec{\boldsymbol{p}}_{\boldsymbol{2}}$.

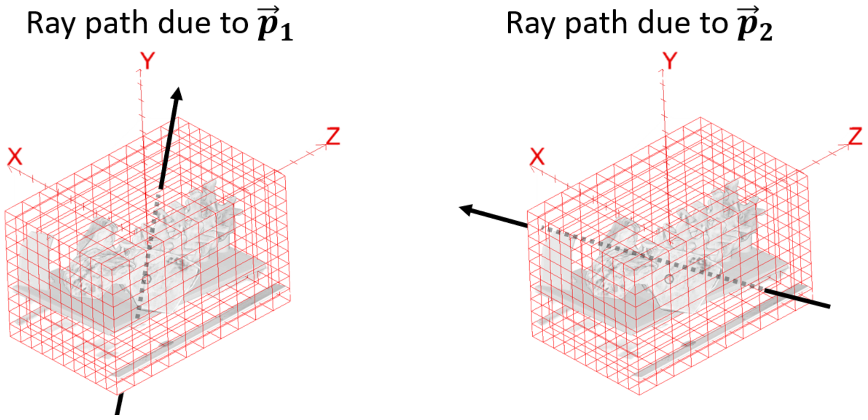

Figure 5-31. Ray path dependency with $\vec{p}$

Bearing that in mind, the performance times are executed testing extreme cases, i.e., using ray-paths that go through the minimum and the maximum number of voxels of the CT volume. The minimum case is achieved when no rotation is applied as the ray-path goes in the negative Y-axis direction. The maximum case can be approximated with a combination of a 45-degree-rotation around the Z- and X-axis. See Figure 5-32.

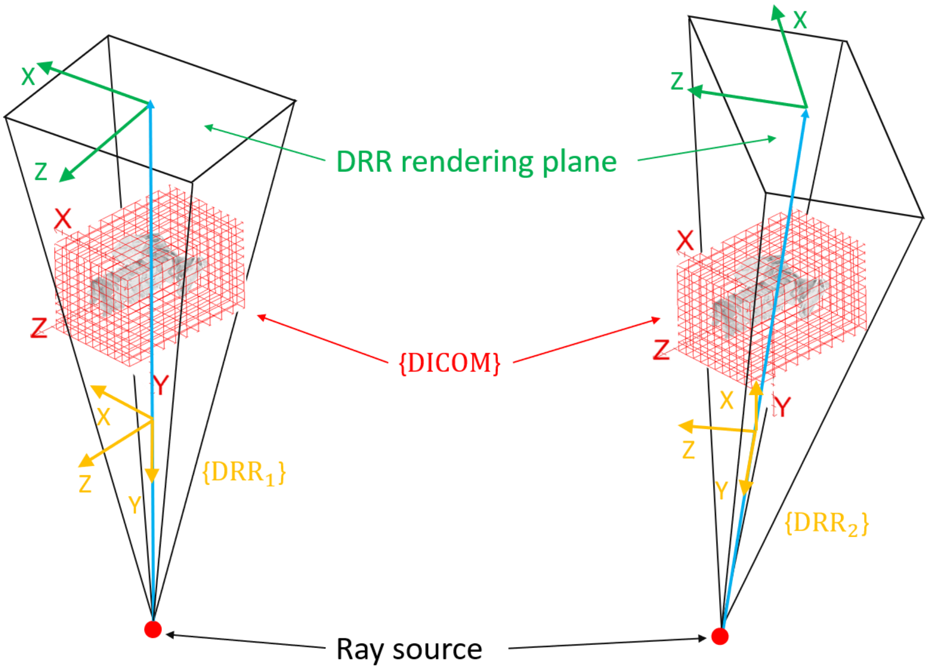

Figure 5-32. Ray-casting path evaluating the minimum and maximum number of voxels

It is stated in section 5.2.5 that the registration procedure uses X-ray images of 568x568px with a pixel spacing of 0.365mm/px. Consequently, the DRR images, which are compared with the X-ray images, are generated with the same resolution and pixel spacing. The results of the performance test can be seen in Table 5-14.

Table 5-14. Ground-truth DRR module performance

Test	$\vec{p}$						Time [s]
	t_x [mm]	t_y [mm]	t_z [mm]	r_x [°]	r_y [°]	r_z [°]	
1	0	0	0	0	0	0	16.12
2	20	20	20	45	0	0	20.14
3	50	50	0	0	0	45	21.66
4	10	20	30	45	0	45	32.85

The results of Table 5-14 indicate that a single DRR image takes around 16 seconds in the best-case scenario, but a time of 32 seconds has to be considered in some specific poses. The 2D/3D registration is achieved using two X-ray images, one AP and one LAT projection. Section 5.2.1 states that each iteration of the 2D/3D registration requires creating twelve DRR images, yet the registration concept is extended in section 5.2.6 by using two X-ray images. This registration with two simultaneous X-ray images uses an additional DRR module, one for each X-ray image. It means that a total of 24 DRR images are required for an iteration, making a time of 384 seconds per iteration. Without considering the total time for a registration, it can be seen that the ground-truth module is impractical for a real scenario; therefore, the DRR module has to be implemented more efficiently while keeping the same characteristics of the rendered image.

5.6.2. DRR Module Implementation

The starting point of the DRR implementation is the Siddon-Jacobs algorithm, described in section 4.7.1, which establishes the line integral of the ray-path as the base for the DRR module. The Siddon-Jacobs algorithm, defined by equations *(34)* and *(35)*, requires an initial and a final point for the ray-path. Every DRR image must calculate a ray-path for each pixel in the rendering plane. The CT volume and the rendering plane are related by $\vec{p}$.

The DRR calculations are split first in the three rotations and afterward in the space translation. It is worth mentioning that the rotations form a set of Euler ZXY rotations. Setting the rotation around the X-axis in the second position is done to avoid a gimbal lock scenario as the 90° rotation in the X-axis is hardly possible to achieve without a collision between the C-arm and patient. See Figure 5-33.

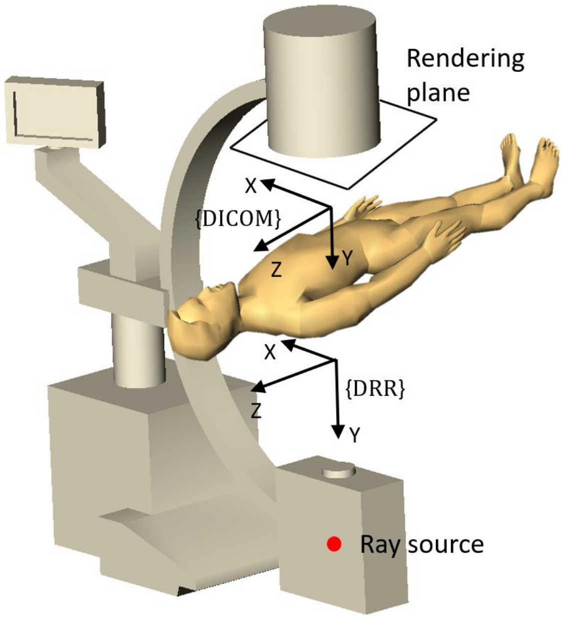

Figure 5-33. ${}^{DRR}\boldsymbol{T}_{DICOM}$ visualization

It is convenient to rotate around the isocenter of the volume instead of the origin of the volume to avoid translations of its origin due to rotations. See Figure 5-34.

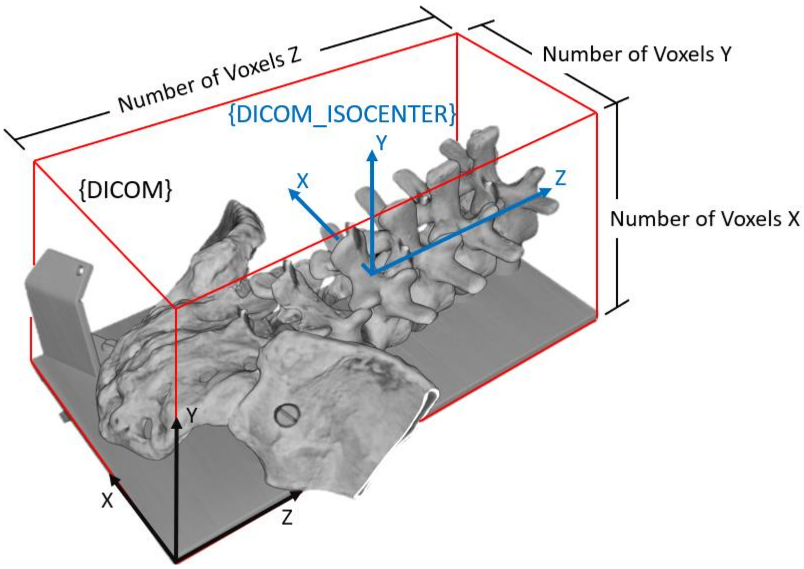

Figure 5-34. Determination of the DICOM Isocenter

Although $\vec{\boldsymbol{p}}$ is equivalent to ${}^{DRR}\boldsymbol{T}_{DICOM}$, it simplifies the calculations if the DRR is computed using the transformation ${}^{DRR}\boldsymbol{T}_{DICOM\_ISOCENTER}$. This calculation can be done since ${}^{DICOM}\boldsymbol{T}_{DICOM\_ISOCENTER}$ is a constant, which depends only on the volume size as expressed in (*123*).

$$ {}^{DICOM}\boldsymbol{T}_{DICOM\_ISOCENTER} = \begin{bmatrix} 1 & 0 & 0 & Number\_Voxels\_X/2 \\ 0 & 1 & 0 & Number\_Voxels\_Y/2 \\ 0 & 0 & 1 & Number\_Voxels\_Z/2 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{123} $$

The given pose to the DRR module is ${}^{DRR}\boldsymbol{T}_{DICOM}$, but it is transformed to ${}^{DRR}\boldsymbol{T}_{DICOM\_ISOCENTER}$ as:

$$ {}^{DRR}\boldsymbol{T}_{DICOM\_ISOCENTER} = {}^{DRR}\boldsymbol{T}_{DICOM} \cdot {}^{DICOM}\boldsymbol{T}_{DICOM\_ISOCENTER} \quad (124) $$

Pixel-Step Method

A critical parameter in the DRR module is the source-detector distance, also known as the C-arm focal length, f, in section 4.5. In the DRR module, the origin is located equidistance along the Y-axis between the source and the detector. The ray-source is represented by point $\boldsymbol{s} = [0, f/2\ , 0]$, and the detector is represented by three points $\boldsymbol{a}$, $\boldsymbol{b}$, and $\boldsymbol{c}$, which defined three corners of the detector. See Figure 5-35. As $\boldsymbol{s}$, $\boldsymbol{a}$, $\boldsymbol{b}$, and $\boldsymbol{c}$ are described by the DRR frame, they can be written as ${}^{DRR}\boldsymbol{p}_s$, ${}^{DRR}\boldsymbol{p}_a$, ${}^{DRR}\boldsymbol{p}_b$, and ${}^{DRR}\boldsymbol{p}_c$, representing the corners of the resulted image in $\mathbb{R}^3$.

As the output image size and pixel spacing are inputs of the DRR module, they are used for computing ${}^{DRR}\boldsymbol{p}_a$, ${}^{DRR}\boldsymbol{p}_b$, and ${}^{DRR}\boldsymbol{p}_c$ as follows:

$$ \begin{aligned} {}^{DRR}\boldsymbol{p}_a &= [-ImageSizeX \cdot pixelSpacing/2, -f/2, ImageSizeY \cdot pixelSpacing/2]^{\boldsymbol{T}} \\ {}^{DRR}\boldsymbol{p}_b &= [ImageSizeX \cdot pixelSpacing/2, -f/2, ImageSizeY \cdot pixelSpacing/2]^{\boldsymbol{T}} \\ {}^{DRR}\boldsymbol{p}_c &= [ImageSizeX \cdot pixelSpacing/2, -f/2, -ImageSizeY \cdot pixelSpacing/2]^{\boldsymbol{T}} \end{aligned} \quad (125) $$

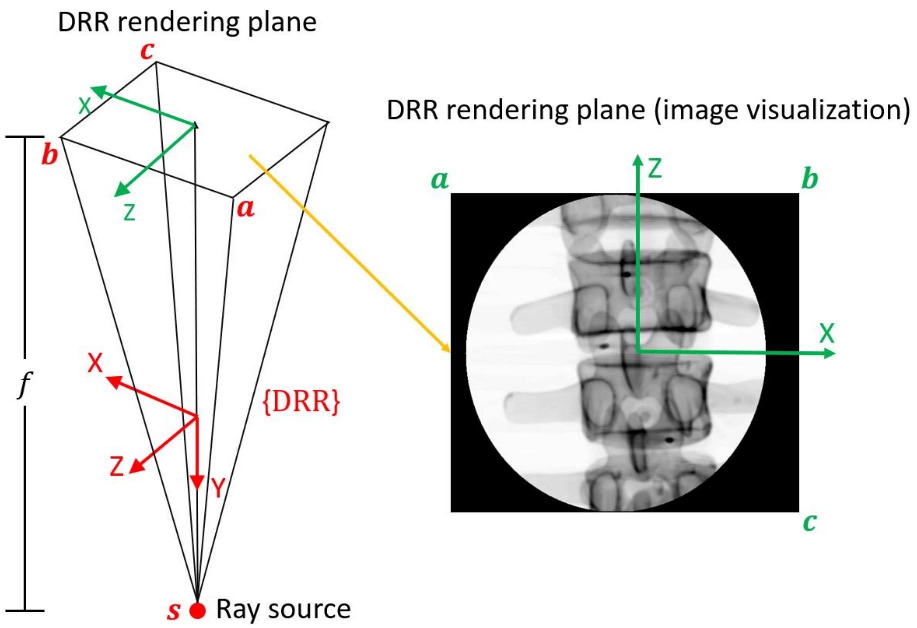

Figure 5-35. DRR reference frames representation

The points ${}^{DRR}\boldsymbol{p}_s$, ${}^{DRR}\boldsymbol{p}_a$, ${}^{DRR}\boldsymbol{p}_b$, and ${}^{DRR}\boldsymbol{p}_c$ are the elements, where the transformation ${}^{DRR}\boldsymbol{T}_{DICOM\_ISOCENTER}$ is applied. Once the new position of those points is achieved, i.e., ${}^{DICOM}\boldsymbol{p}_s$, ${}^{DICOM}\boldsymbol{p}_a$, ${}^{DICOM}\boldsymbol{p}_b$, and ${}^{DICOM}\boldsymbol{p}_c$, each pixel of the image can be found by adding offsets of pixel spacing sizes in X- and Z-axis in {DICOM} [175]. The pixel (n, m) of the DRR image can be calculated using the equation (*49*) and (*50*) with the initial point, ${}^{DICOM}\boldsymbol{p}_s$, and the final point ${}^{DICOM}\boldsymbol{p}_{\mathrm{PIXEL\_}(n,m)}$. The above procedure is illustrated in Figure 5-36.

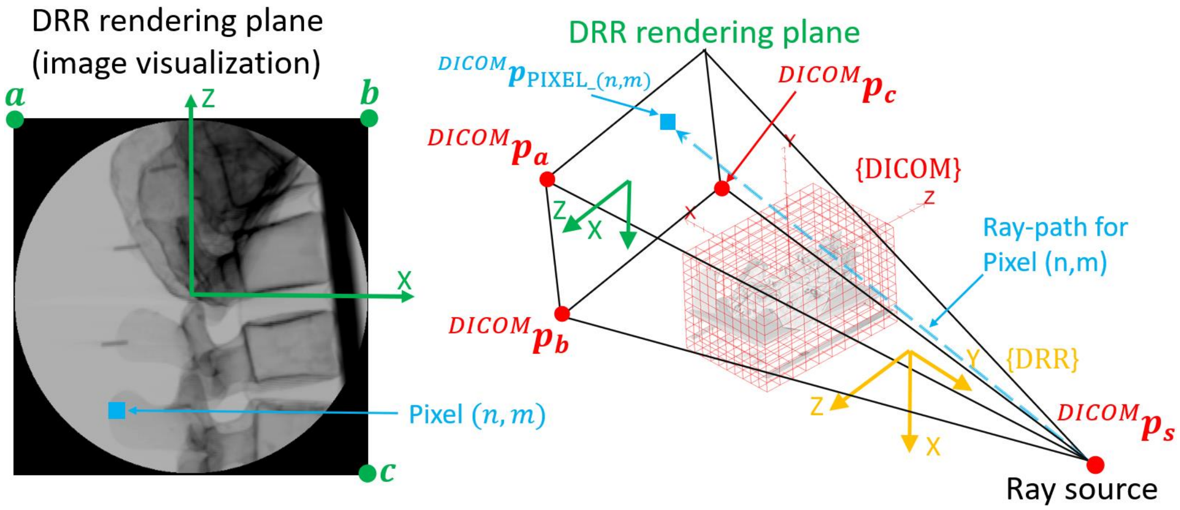

Figure 5-36. DRR pixel rendering process

A DRR image is rendered after the $N \times M$ pixels of the image are calculated. In this particular case, the created images consist of 568x568 pixels, which compute 322624-line integrals, one per pixel. In the center of Figure 5-37, it can be seen a DRR image created with the own implementation; on the left side, it can be seen the DRR image from the ground-truth module using the same $\vec{\boldsymbol{p}}$. Both images have the grayscale range [0-255]. The pixel-wise absolute difference between both images is calculated and displayed on the right side of Figure 5-37. The darker the resulted image, the more similar the two DRR images are. It can be noticed that consistent gray background color and some grayscale patches on the body segment are visible. The patches are due to approximation errors and edge handling between implementations. The whitest spots visible in the pixel-wise absolute difference image have a grayscale value of 11, while the darkest spots have a grayscale value of 4. In the visualization of the difference image, the contrast was increased to the maximum; otherwise, it would not have been possible to visualize the difference as the grayscale range is [4-11]. This reduced grayscale range is a good indicator of the similarity between the two approaches.

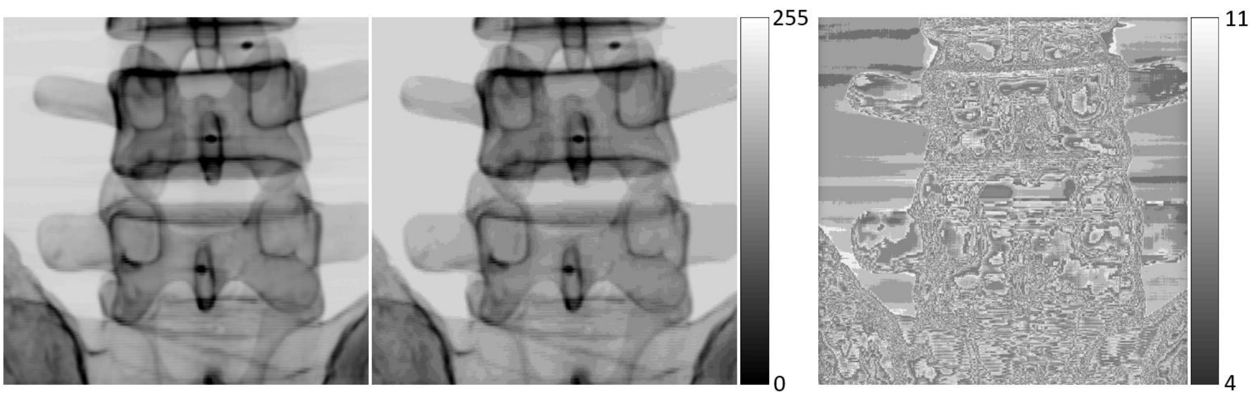

Figure 5-37. DRR image rendering: Left ground truth, center own implementation. Pixel-wise absolute difference between DRR images on the right

Since render quality between the two approaches is equivalent, the evaluation of the implementation can be focused on the rendering time.

The rendering time measurement is carried out following the same poses in Table 5-14, image dimension of 568x568 pixels, and pixel spacing of 0.365 px/mm. The results can be seen in Table 5-15.

Table 5-15. Own DRR module performance

Test	$\vec{p}$						Time [s]	Speed improvement with respect to the ground-truth. [n-folds]
	t_x [mm]	t_y [mm]	t_z [mm]	r_x [°]	r_y [°]	r_z [°]		
1	0	0	0	0	0	0	1.71	9.43
2	20	20	20	45	0	0	2.41	8.36
3	50	50	0	0	0	45	2.94	7.67
4	10	20	30	45	0	45	4.01	8.19
			Average speed improvement [n-folds]:					8.41

It can be noticed that the implemented DRR is about 8.4 times faster than the ground-truth module. It is a significant improvement just based on the pixel-step method and the implementation efficiency. However, considering the rendering times applied to the 2D/3D registration, one iteration would take around 41 seconds in the best case. The new implementation does not reach an execution time to be practical for a real-scenario; therefore, further optimizations must be considered.

5.6.3. DRR Module Implementation with Parallel Computing

The Siddon-Jacobs algorithm was created initially for serial computing, but its working principle allows parallelization, which leads to an increment in the computing speed. In the implementation in section 5.6.2, for every pixel, the required initial and final points for the ray-casting algorithm are calculated using the pixel-step method, and then the Siddon-Jacobs algorithm is applied by calculating the line integral. This procedure uses a serial approach, which means that the next pixel in line starts once the previous pixel is calculated. Using a parallel approach, several pixels can be calculated simultaneously as the calculation of one pixel is independent of any other pixel.

One approach consists of using multi-threading. Each thread can be executed simultaneously and independently in the available cores inside the computer's central processing unit (CPU). This work was developed in a computer with an Intel i7-8800K processor, which contains six cores, two logic processors per core, and can execute one thread per logic process, i.e., it can run up to 12 threads. The proper implementation could lead to a maximum theoretical further improvement of 12 times the current speed, which means a DRR image could be rendered in around 230ms and a registration iteration executed in about 5.5 seconds. Considering synchronization and preparations for splitting the calculation in multiple threads, the final time would not reach the 12 times improvement, so a registration iteration would not reach the 5.5 seconds mark.

Another approach is to use the computational power of the graphics processing unit (GPU). Intrinsically, the GPU is designed for high-speed graphics, which are inherently parallel [176]. While CPUs focus on low latency, GPUs focus on high throughput [177]. That means the CPU is faster than the GPU to accomplish a task, but the GPU is faster to process thousands of tasks due to its parallel capabilities. With the Compute Unified Device Architecture (CUDA) release by NVIDIA in 2006, GPUs were brought into the general-purpose processing applications [178].

CUDA is a parallel computing platform, which simplifies the programming process for parallel computing in the GPU. CUDA allows defining C++ functions, called kernels, which are executed N times in parallel by N different CUDA threads [178]. In this work, an Nvidia GTX 1080 graphics card was used, which can execute 16 threads per CUDA core, and it has 2560 CUDA cores. Consequently, the GPU can execute up to 40960 threads simultaneously. Since the GPU processor frequency, 1.57Ghz is 2.4 times slower than the used CPU, 3.7Ghz, the computation of a single-pixel of the DRR image takes longer using a GPU thread. However, the vast number of simultaneous pixels the GPU can process is enough to increase the overall execution time, as Table 5-16 shows.

Table 5-16. Parallel computing DRR module performance

Test	$\vec{\boldsymbol{p}}$						Time [ms]	Speed improvement with respect to the	
	t_x [mm]	t_y [mm]	t_z [mm]	r_x [°]	r_y [°]	r_z [°]		ground-truth. [n-folds]	Serial implementation [n-folds]
1	0	0	0	0	0	0	4.71	3422.5	363.1
2	20	20	20	45	0	0	5.98	3367.9	403
3	50	50	0	0	0	45	6.57	3296.8	447.5
4	10	20	30	45	0	45	22.80	1440.8	175.9
			Average speed improvement [n-folds]:					2882	347.4

The Siddon-Jacobs algorithm and the pixel-step method do not change by the parallel implementation, only the functions invocation. That means the visual results remain the same as with the serial implementation. Table 5-16 shows the boost in the performance by the parallel computing implementation of the DRR. Now the average rendering times are around 10ms, showing an improvement of 347 times compared with the serial implementation and 2882 times with the ground-truth, more than three orders of magnitude.

These rendering times in the 2D/3D registration context indicate that one iteration can be done in around 240ms and a 20-iteration registration in 4.8 seconds. This implementation fulfills the overall execution time for the 2D/3D registration procedure in a real scenario.

5.6.4. Discussion and Conclusions

The DRR module evaluation is carried out using a CT-scan of 512x512x442 voxels and creating a DRR image of 568x568 pixels with a pixel spacing of 0.365mm/pixel. The ray-path is critical when it crosses many voxels, so the evaluation is done using a ${}^{DRR}\boldsymbol{T}_{DICOM}$ that makes the ray cross the diagonal of the volume cube.

A DRR module in the Medical framework and visualization (MeVisLab) was selected as the ground-truth DRR. The evaluation of the module gave an average DRR image rendering time of 22 seconds.

The Siddon-Jacobs algorithm is used as the ray-casting method to render pixels to implement a DRR for the 2D/3D registration. Six DoF are further added to the pixel-step method, an approach based on transforming four points. One point is the ray-source; the other three are the corners of the image detector. The position of any pixel on the transformed detector is calculated using the novel pixel-step method.

With the DRR serial implementation using the pixel-step method, the average DRR rendering time is reduced to 2.8 seconds, which consists of an 8.4-fold improvement compared to the ground truth module. Further optimization of the DRR implementation is carried out using parallel computing. Using the GPU and the CUDA libraries for C++, the Siddon-Jacobs algorithm is adapted to run as a CUDA kernel function. A GPU thread can execute a kernel function, and in the case of the available Nvidia GTX 1080 card, it can run up to 40.960 threads in parallel. It has to be considered that the GPU frequency, 1.57Ghz, is 2.4 times slower than the used CPU, Intel i7-8700k at 3.7Ghz.

The parallel computing implementation reaches an average execution time of 10ms, which is 347 times faster than the serial implementation and almost 3000 times faster than the ground-truth DRR module, more than three orders of magnitude faster. With the runtime of the parallel DRR, an iteration of the 2D/3D registration is done in around 240ms and a complete registration of 20 iterations in 4.8 seconds. These times are acceptable for the 2D/3D registration in a real scenario.

5.7. Manual Selection of the Initial Pose for the Registration Procedure

The implemented intensity-based 2D/3D registration is an iterative optimization process, which requires an initial pose for starting the optimization, as explained in sections 4.7 and 5.2. Since the optimization result is the pose ${}^{DRR}\boldsymbol{T}_{DICOM}$, then the initial pose is also a ${}^{DRR}\boldsymbol{T}_{DICOM_0}$. The transformation matrix ${}^{DRR}\boldsymbol{T}_{DICOM}$ expresses six DoF, which are gathered in a vector of six elements, i.e., $\vec{\boldsymbol{p}} \in \mathbb{R}^6$as defined in equation (*47*). Regardless of the used notation, the initial pose for the optimization is a multidimensional array, representing a challenge to be inputted manually into the 2D/3D registration. On top of that, the final users for this application are non-technical users, health workers, who should be unaware of the mathematical background involved in the registration process. Finally, it is worth mentioning that the selection of the initial pose influences the registration accuracy as the selection of the initial guess affects the results of an optimization procedure. See section 4.7.3.

In this section, an innovative initial pose-selector is developed using only graphical interaction based on the nature of the inputs used for the registration, e.g., 2D and 3D medical imaging. This initial pose-selector gives the advantage that the technical aspects are not disclosed to the final user, and a close-to-the-registration pose is graphically selected by physicians, who have a well-trained eye for manipulating modalities such as CT-scans and X-rays. The idea behind the graphical pose-selector module consists of generating a DRR image from the current view of the DICOM volume. The given pose to the registration process, ${}^{DRR}\boldsymbol{T}_{DICOM_0}$, is taken from the initial pose generator.

5.7.1. Reference Frames of the 3D and 2D Viewers

The visualization of the 3D volume is carried out using a MeVisLab native module called examiner viewer. It allows a user to interact with a 3D object using the mouse actions, i.e., scroll, left-, and right-click. The most important feature of the examiner viewer is that it reports the 3D object pose with respect to the examiner viewer reference frame, {ExaminerViewer}. As the used 3D object is a DICOM volume, the reported pose is ${}^{ExaminerViewer}\boldsymbol{T}_{DICOM}$. See Figure 5-38.

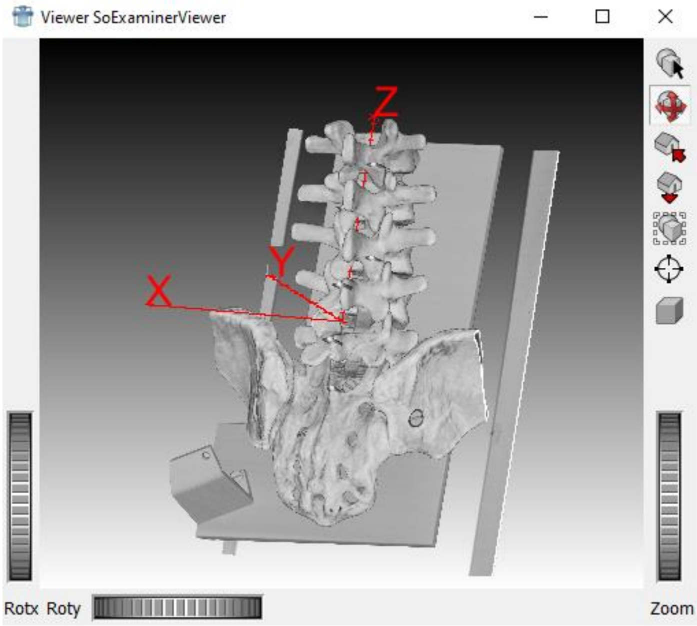

Figure 5-38. 3D volume rendered in a MeVisLab examiner viewer

The {ExaminerViewer} is fixed to the rendering window. The manual pose-selector consists of creating a DRR image from the examiner viewer's current scene; mathematically, it can be expressed as ${}^{ExaminerViewer}\boldsymbol{T}_{DRR}$. The transformation ${}^{ExaminerViewer}\boldsymbol{T}_{DRR}$ associates the examiner viewer view with the DRR view. The first step to find the previous transformation requires determining the examiner viewer reference frame. It is done by bringing the examiner viewer to a zero-rotation and zero-translation state. At that moment, the {DICOM} is aligned with the {ExaminerViewer}. It can be seen on the left side of Figure 5-39 the {DICOM} within the examiner viewer window. That means, {ExaminerViewer} is the fixed frame located on the window center, drawing on the right side of Figure 5-39.

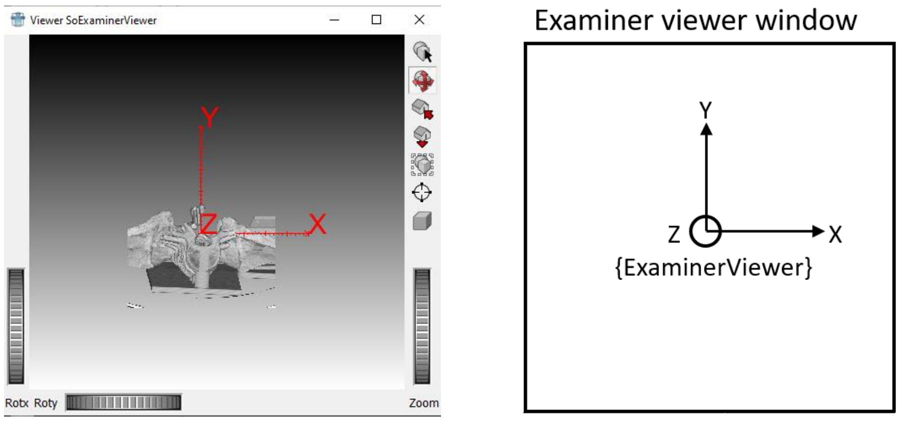

Figure 5-39. Description of the examiner viewer reference frame

The Y-axis of {ExaminerViewer} points up, and the X-axis goes from left to right while the Z-axis pops out of the examiner viewer window.

In section 5.6, the DRR module is developed in detail. The image reference frame of the DRR module is explained in Figure 5-35. Making an analogy with the {ExaminerViewer}, on the right side of Figure 5-39, the {DRR} is shown in Figure 5-40.

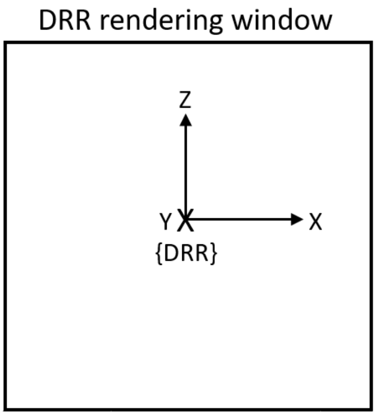

Figure 5-40. View of the DICOM with Zero-Rotation from the DRR Module

In this case, the DRR window renders an image on the Z-X plane with the Y-axis entering onto the picture. Both window frames, i.e., {ExaminerViewer} and {DRR} stay put. That means the transformation ${}^{ExaminerViewer}\boldsymbol{T}_{DRR}$ is a constant. In Figure 5-41, both reference frames can be seen side by side. It can be noticed that the {ExaminerViewer} must rotate negative 90 degrees in its X-axis to reach the same axes distribution as the DRR window. It is assumed that both windows origins do not change, so no translation has to be added.

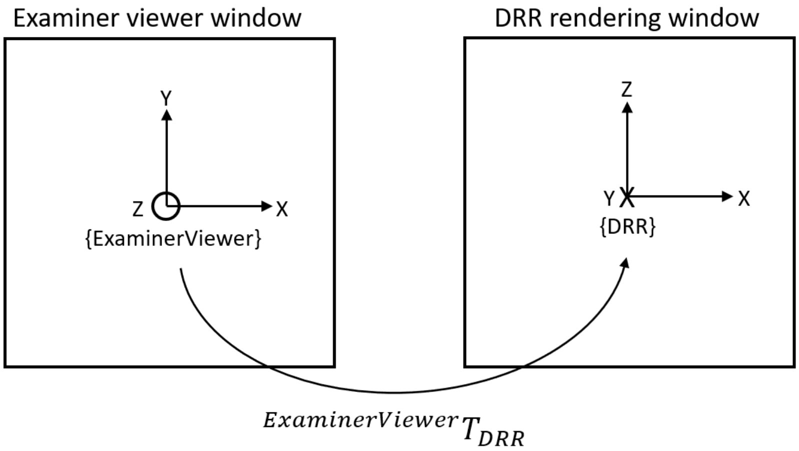

Figure 5-41. Comparison of the DICOM Reference Frame in the Examiner Viewer and the DRR Module

Bearing this in mind, $^{ExaminerViewer}\boldsymbol{T}_{DRR}$ can be found as:

$$^{ExaminerViewer}\boldsymbol{T}_{DRR} = \begin{bmatrix} Rot\{x, -90°\} & \vec{\mathbf{0}} \\ \vec{\mathbf{0}}^T & 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & -1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{126}$$

5.7.2. Transformation Between the 3D-Viewer and DRR Module

The set of reference frames involved in the manual pose selection are depicted in Figure 5-42.

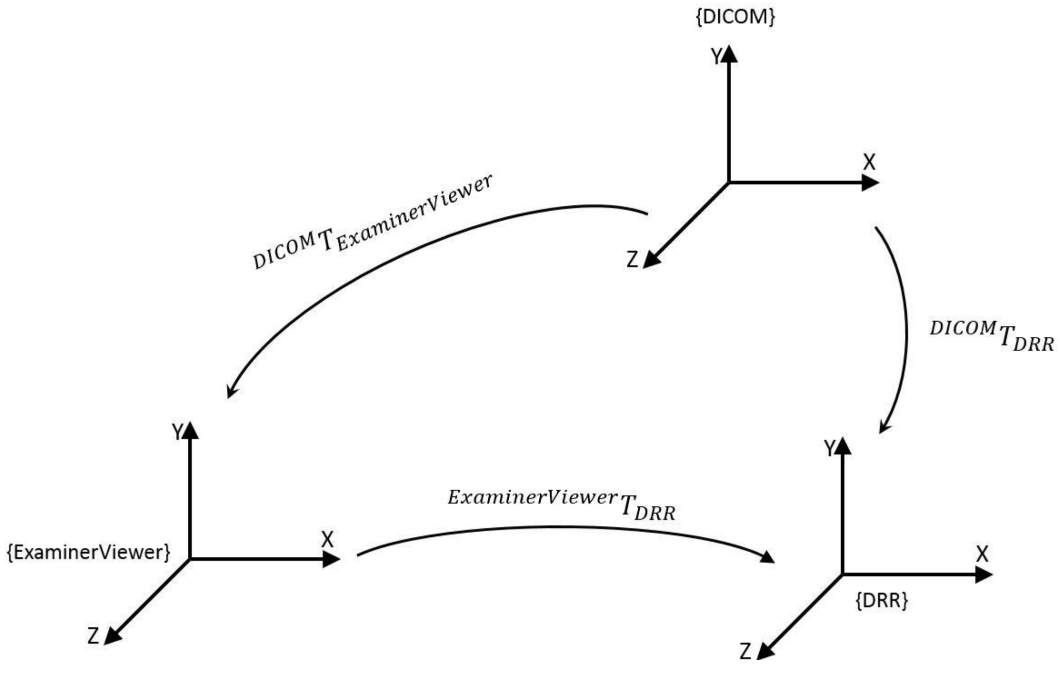

Figure 5-42. Transformation for Making the DRR Image from the examiner viewer View

It is known from section 5.6 that the input pose of the DRR module is $^{DRR}\boldsymbol{T}_{DICOM}$. This pose is found using the reference frames in Figure 5-42 as:

$$^{DRR}\boldsymbol{T}_{DICOM} = (^{ExaminerViewer}\boldsymbol{T}_{DRR})^{-1} \cdot {}^{ExaminerViewer}\boldsymbol{T}_{DICOM} \tag{127}$$

Replacing the constant found in (*126*) into (*127*), $^{DRR}\boldsymbol{T}_{DICOM}$ can be expressed as:

$$^{DRR}\boldsymbol{T}_{DICOM} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & -1 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \cdot {}^{ExaminerViewer}\boldsymbol{T}_{DICOM} \tag{128}$$

Equation (*128*) gives the transformation to create DRR images from the same perspective as the examiner viewer.

Interpretation of the Depth Parameter in the Examiner Viewer

The examiner viewer presents an additional challenge. Changing the DICOM depth visualization is done by a zoom feature, which does not change the reported coordinates, i.e., $^{ExaminerViewer}\boldsymbol{T}_{DICOM}$. The zoom feature can be seen as an additional reference frame, {ExaminerViewerZoom}, that gets closer or farther from the {ExaminerViewer}. The reported zoom value, $H \in \mathbb{R}^{+}$, represents the closeness of the {ExaminerViewerZoom} from the examiner viewer center. From another perspective, H moves the rendering plane in the Z-Axis of the window view. A graphical explanation can be seen in Figure 5-43.

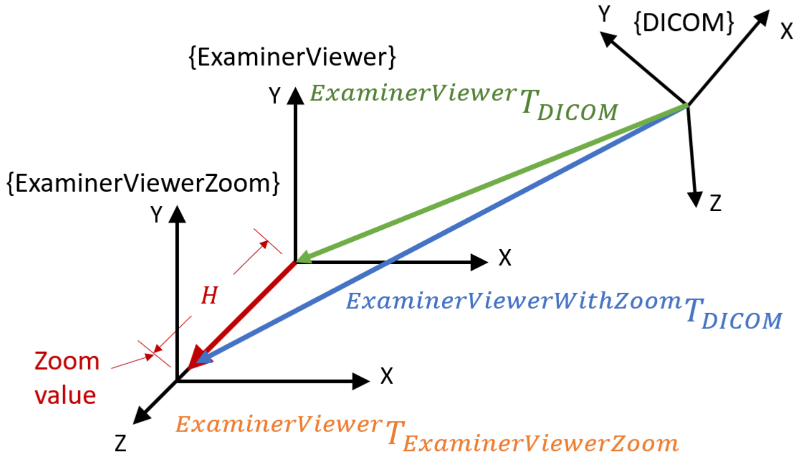

Figure 5-43. Conversion from the Window Zoom to the ExaminerViewer Frame

H is incorporated to find feedback from the examiner viewer that gives six DoF. Additionally, the reference frame that gives the full pose is the {ExaminerViewerZoom} instead of the {ExaminerViewer}. That means equation (*128*) is updated to:

$$^{DRR}\boldsymbol{T}_{DICOM} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & -1 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \cdot {}^{ExaminerViewerZoom}\boldsymbol{T}_{DICOM} \tag{129}$$

As H is also reported from the examiner viewer, the transformation $^{ExaminerViewer}\boldsymbol{T}_{ExaminerViewerZoom}$ is built with no rotation, and H is included as the Z-component of the translation:

$$^{ExaminerViewer}\boldsymbol{T}_{ExaminerViewerZoom} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & H \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{130}$$

Now ${}^{ExaminerViewerZoom}\boldsymbol{T}_{DICOM}$ can be calculated using the transformation chain depicted in Figure 5-43 as:

$$ {}^{ExaminerViewerZoom}\boldsymbol{T}_{DICOM} = \left({}^{ExaminerViewer}\boldsymbol{T}_{ExaminerViewerZoom}\right)^{-1} \cdot {}^{ExaminerViewer}\boldsymbol{T}_{DICOM} \quad (131) $$

Replacing (*130*) and (*131*) into (*129*), and simplifying:

$$ {}^{DRR}\boldsymbol{T}_{DICOM} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & -1 & H \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \cdot {}^{ExaminerViewer}\boldsymbol{T}_{DICOM} \quad (132) $$

Equation (*132*) gives the transformation to create DRR images from the same perspective as the examiner viewer. It uses ${}^{ExaminerViewer}\boldsymbol{T}_{DICOM}$ and H, which are values provided by the examiner viewer.

5.7.3. Results

In Figure 5-44, Figure 5-45, and Figure 5-46, it is possible to see the DICOM volume rendering on the right side and the DRR image automatically created from that perspective.

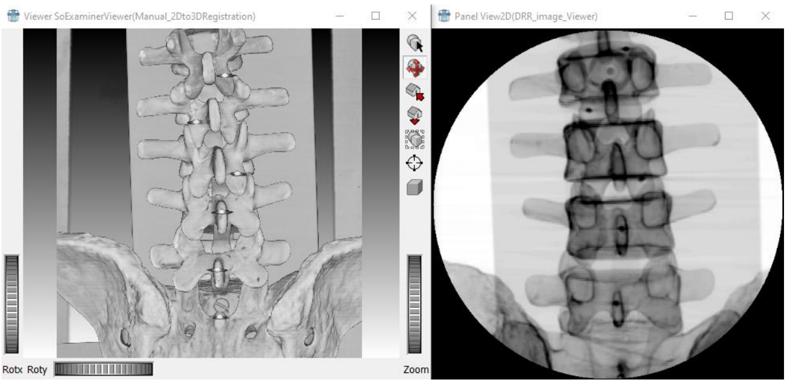

Figure 5-44. DRR image created from the examiner viewer perspective. Example 1

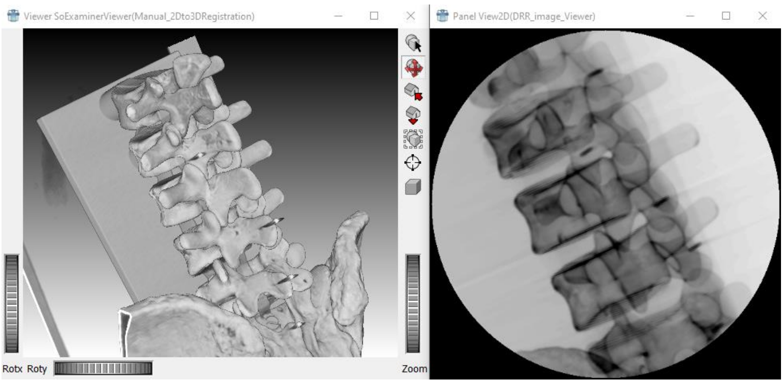

Figure 5-45. DRR image created from the examiner viewer perspective. Example 2

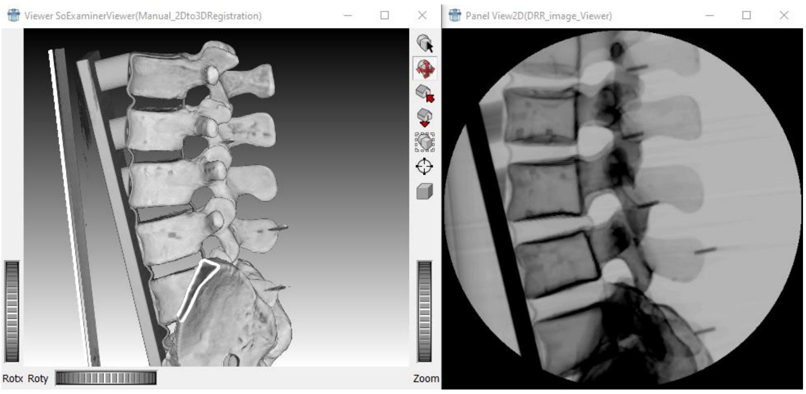

Figure 5-46. DRR image created from the examiner viewer perspective. Example 3

In the 2D/3D registration interface, the user is aided by the pixel-wise absolute difference between the generated DRR image and the X-ray used for the registration. That means a user can also see the overlapping between the DRR and the X-ray image as guidance to create an initial pose to the registration algorithm. In Figure 5-47 and Figure 5-48, the DICOM volume rendering is seen on the left side. The actual X-ray image is shown on the upper central part, and the DRR image is seen on the bottom center. On the right, the pixel-wise absolute difference of the two central images is rendered.

Figure 5-47. Initial pose selection in the 2D/3D registration user interface. Example 1

Figure 5-48. Initial pose selection in the 2D/3D registration user interface. Example 2

5.7.4. Discussion and Conclusions

The initial pose selection of the 2D/3D registration is achieved using the native 3D visualizer of the used medical framework, MeVisLab. The 3D visualizer, called examiner viewer, reports the pose form {ExaminerViewer} to {DICOM}, i.e., ${}^{ExaminerViewer}\boldsymbol{T}_{DICOM}$, which is a useful functionality exploited to create the graphical initial pose selection.

The initial pose is taken as the pose that creates a DRR image with the same perspective as the examiner viewer is rendering the 3D volume. In other words, the initial pose is the transformation ${}^{DRR}\boldsymbol{T}_{DICOM}$, created from ${}^{ExaminerViewer}\boldsymbol{T}_{DICOM}$.

The graphical manipulation of the DICOM is conveniently presented as the pixel-wise absolute difference between the X-ray image to be registered and the DRR image created from the current DICOM view. That final absolute difference helps as a guide for the manual selection of the initial pose.

5.8. Local and Global 2D/3D Registration based on the Region of Interest (ROI)

The benefits of using a ROI in the 2D/3D registration are stated in section 5.2.5, e.g., decreasing the DRR rendering time and filtering undesired elements on the X-ray image. Section 5.5.4 develops the image similarity measurement whose inputs, the DRR and X-ray image, are evaluated only within the ROI boundaries. It is specified that including a ROI in the image similarity decreases the computation time and gives a better similarity estimation, which improves the registration result.

These previous ideas are put together in this section, helping to define the concepts of global and local registration. It is shown that the current 2D/3D registration can be further classified as a local 2D/3D registration. After that, an experiment is carried out, validating the idea of local 2D/3D registration. Finally, an approach to build a quasi-global 2D/3D registration is discussed by an approximation using a piecewise function.

5.8.1. Local 2D/3D Registration

The region of interest is formally defined in section 5.2.5 as two points in the image space forming the main diagonal of a rectangle. There are a couple of advantages from the implementation side to defining a ROI in the 2D/3D registration process. First of all, no every feature in the X-ray image is considered for the registration process, e.g., calculating image similarity measurement. Also, the ROI can be used to reduce the size of the generated DRR image, which influences the rendering time; therefore, the 2D/3D registration can be executed faster thanks to the ROI definition.

The 2D/3D registration requires a 3D volume as an input; in this work, a CT-scan. See section 4.7. CT-scans are taken as pre-operative images, while X-ray images belongs to the intra-operative data category. The definitions of pre- and intra-operative data are elaborated in 4.6.1. One procedural aspect worth considering is the patient positioning during the image modalities acquisition. For the CT-scan, the patient lies his/her back on the operating table, supine position [179], see the left side of Figure 5-49.

On the contrary, the patient is laid down on his/her chest, exposing the patient's back for the spine surgery, prone position. The difference between the patient orientation during the modalities affects the spine shape, as the right side of Figure 5-49 shows. During the CT-scan, the spine tends to flat while keeping the natural curvature during the X-ray. Although the spine changes from a macro view, some rigid components, vertebrae, do not deform. The spine is shaped by flexible deformations of the spine disks, which shifts the relative position among rigid vertebrae.

Courtesy of AKTORmed™

Figure 5-49. Patient positioning during modalities acquisition for spine surgery. CT-scan on the left, X-ray on the right

Trying to register every visible vertebra in an X-ray image with respect to the CT-scan implies that the spine deformations are also considered, and that one transformation represents the registration between the pre- and intra-operative imaging. A registration of this type represents a global 2D/3D registration, but the result is an elastic (non-homogenous) transformation.

From another perspective, rigid 2D/3D registrations could be possible to achieve among rigid elements, i.e., one vertebra in the X-ray image with respect to the same vertebra in the 3D volume. This registration can be achieved without considering deformations as long as a single rigid element is used. The selection of this rigid element can be made using a ROI on the X-ray image. As only one of the several vertebras in the 3D volume is considered, the achieved registration can be perceived as a local 2D/3D registration.

Since the local 2D/3D registration uses a ROI, then the advantages of using ROIs are brought to the local registration. That means the registration is done faster than using the entire image, plus a filtering stage on the X-ray image is implicitly added. Using a ROI also focuses on the image similarity measurement to one specific vertebra, given more chances to have a successful and accurate 2D/3D registration.

To validate and test the local 2D/3D registration, a set of registrations are executed. The setup uses the spine model used in section 5.5. Ten AP and ten LAT X-ray images are captured in different poses. The registrations are made using the combinations of one AP with another LAT images, i.e., 100 combinations.

Additionally, four different ROIs are selected in each of the combinations, where each ROI includes a different lumbar vertebra, L1 to L4. Figure 5-50 illustrates the ROI selection of the L3 vertebra in both AP and LAT projection.

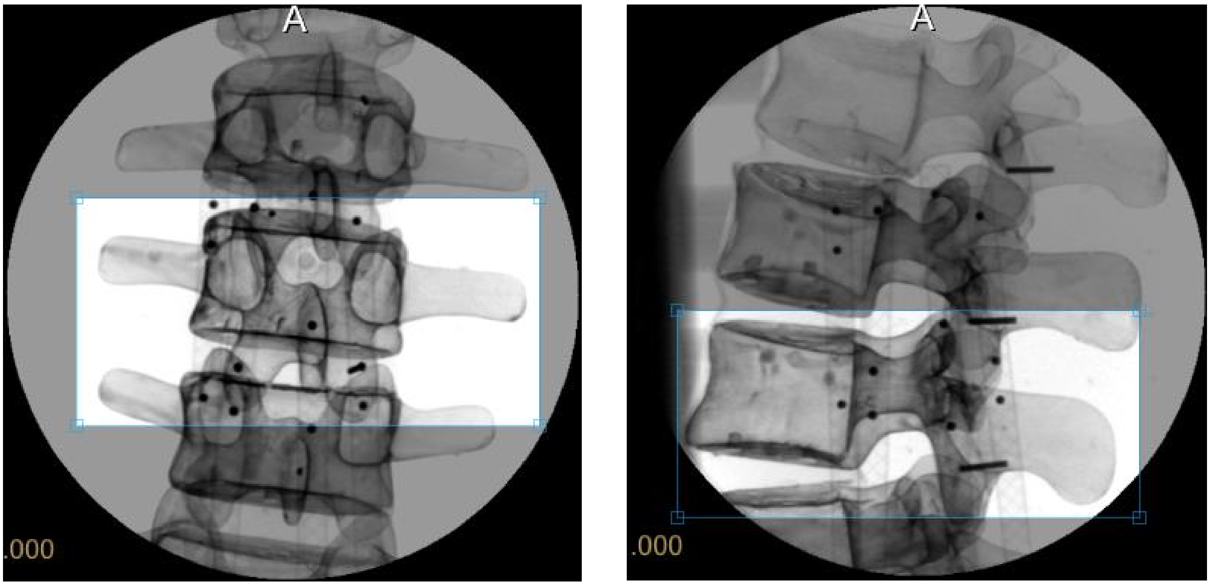

Figure 5-50. ROI selecting the L3 vertebra in DICOM volume. AP projection on the left, and LAT projection on the right

All in all, 500 registrations are executed, 100 registration without ROI and 400 registrations including the ROIs from the vertebrae L1 to L4. For this test, four screws are planned, one in each vertebra. It can be seen in Figure 5-51 the location of each planned implant in the CT-scan of the spine model.

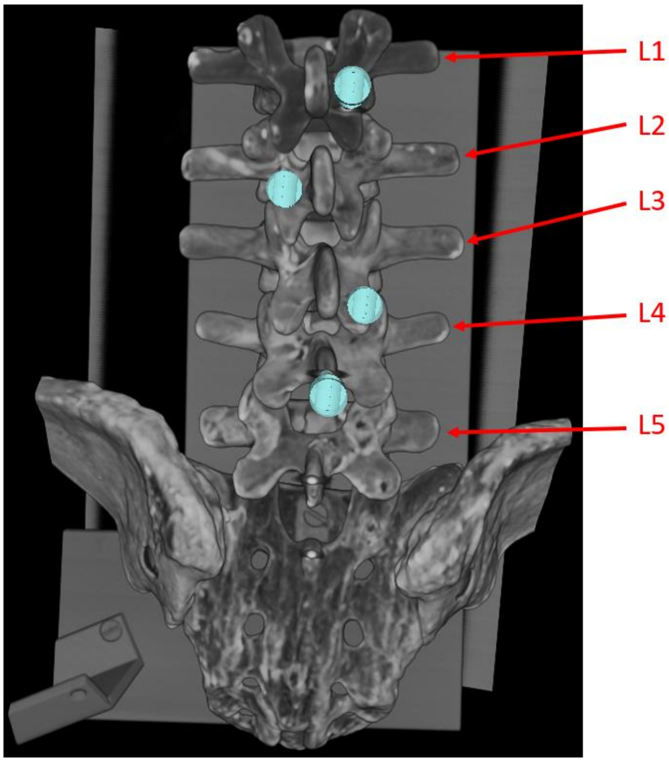

Figure 5-51. DICOM file with planed implants for the ROI test

The registration accuracy is measured using the procedure described in section 5.2.7, equation (*104*). The results are gathered in the rows by the vertebra selected with the ROI, i.e., L1 to L4. Each column shows each implant registration RMS error, which is distributed in different vertebrae, L1 to L4. The results of each row are made by 100 registration and are displayed in Table 5-17.

Table 5-17. Local 2D/3D registration accuracy results

ROI selecting vertebra	RMS error in vertebra			
	L1 [mm]	L2 [mm]	L3 [mm]	L4 [mm]
No ROI	1.10	1.68	1.59	1.81
L1	0.93	1.79	1.79	2.23
L2	0.91	1.50	1.56	2.01
L3	1.05	1.40	1.31	1.60
L4	1.51	1.56	1.20	1.56

It can be seen that the accuracy of the 2D/3D registration is higher when the evaluated screw is included in the ROI, diagonal highlighted, in comparison with the case without ROI, “No ROI” row. In other words, the accuracy of the registration inside the ROI is better than without a ROI. The implant in the vertebra L1 improves from 1.10mm to 0.93mm, the implant in L2 from 1.68mm to 1.5mm, in L3 from 1.59mm to 1.31mm, and the implant in L4 from an RMS error of 1.81mm to 1.56mm. The use of ROIs represents an accuracy improvement of around 0.2mm in contrast without using ROI. The previous observations support the initial assumption of a local 2D/3D registration, where a zone in the image is registered with high accuracy, which is defined by a ROI.

5.8.2. An Approach to Global 2D/3D Registration

To avoid including deformation and non-linear transformations, the global 2D/3D registration is created as the union of local registrations. Analogous to the vertebra selection in a 2D image using a ROI, the vertebrae in the 3D volume are split using segmentation. Each segmented vertebra in the volume is registered with respect to the corresponding vertebra in the 2D image framed by the ROI.

DICOM Segmentation

Analog to the ROI in a 2D image defined by a square, the segmentation of the volume is done using a cube. Although a cube has eight vertices, it can be defined by two points. Let two points $\mathbf{s}_{x1}$ and $\mathbf{s}_{x2}$ be defined as:

$$\{(x_{x1}, y_{x1}, z_{x1}), (x_{x2}, y_{x2}, z_{x2}) \in \mathbb{Z} \mid \boldsymbol{s}_{x1} = (x_{x1}, y_{x1}, z_{x1}), \boldsymbol{s}_{x2} = (x_{x2}, y_{x2}, z_{x2})\} \tag{133}$$

The segmentation volume is defined by $\boldsymbol{s}_{x1}$ and $\boldsymbol{s}_{x2}$, representing the end-points of the cube main-diagonal, as Figure 5-52 shows.

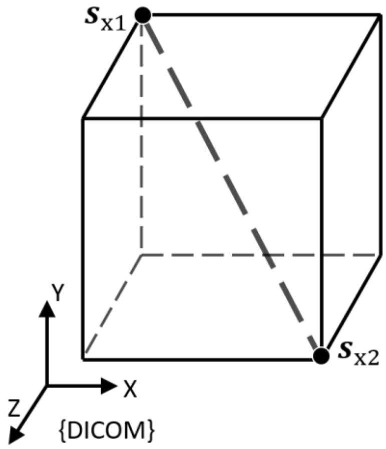

Figure 5-52. Two points describing the main diagonal of a cube

The points $\boldsymbol{s}_{x1}$ and $\boldsymbol{s}_{x2}$ are expressed in {DICOM}, i.e., ${}^{DICOM}\boldsymbol{p}_{s_{x1}}$ and ${}^{DICOM}\boldsymbol{p}_{s_{x2}}$. The cube defined by the $\boldsymbol{s}_{x1}$ and $\boldsymbol{s}_{x2}$ describes a new reference frame named {SegmentedVertebraLx}, which contains only the voxels included within the cube. This new 3D volume is used for the 2D/3D registration as it only contains the vertebra Lx. The transformation ${}^{DICOM}\boldsymbol{T}_{SegementedVertebraLx}$ can be deduced from the points $\boldsymbol{s}_{x1}$ and $\boldsymbol{s}_{x2}$. As explained in section 5.6, ${}^{DRR}\boldsymbol{T}_{DICOM}$ considers the {DICOM} origin to be located at the DICOM volume center; therefore, the transformation ${}^{DICOM\_ISOCENTER}\boldsymbol{T}_{SegementedVertebraLx\_ISOCENTER}$ must be considered from the segmentation process. It can be seen in Figure 5-53 the transformations involved in the segmentation process using a DICOM spine model and segmenting the vertebra L3.

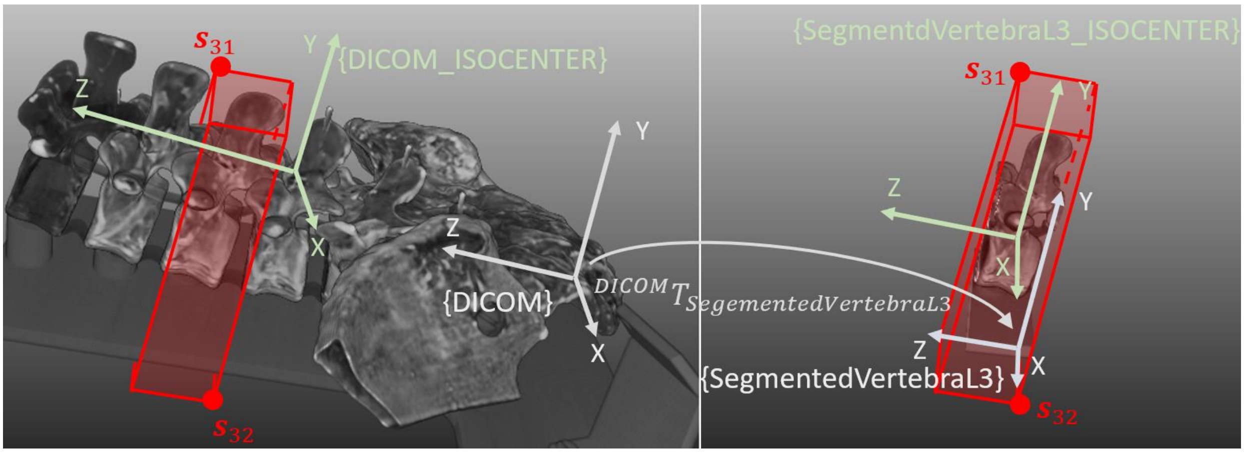

Figure 5-53. DICOM to Segmented vertebra Lx volume center

Using equation (122), it is straightforward to find ${}^{DICOM}\boldsymbol{T}_{DICOM\_ISOCENTER}$ and ${}^{SegmentedVertebraLx}\boldsymbol{T}_{SegmentedVertebraLx\_ISOCENTER}$, which remain as constants. ${}^{DICOM\_ISOCENTER}\boldsymbol{T}_{SegementedVertebraLx\_ISOCENTER}$ can be calculated using as follows:

$$
{}^{DICOM\_ISOCENTER}\boldsymbol{T}_{SegementedVertebraLx\_ISOCENTER} = {}^{DICOM\_ISOCENTER}\boldsymbol{T}_{DICOM} \cdot {}^{DICOM}\boldsymbol{T}_{SegementedVertebraLx} \cdot {}^{SegementedVertebraLx}\boldsymbol{T}_{SegementedVertebraLx\_ISOCENTER} \tag{134}
$$

In the remaining of this section, the used transformation will be ${}^{DICOM}\boldsymbol{T}_{SegementedVertebraLx}$, which has to be converted to and from ${}^{DICOM\_ISOCENTER}\boldsymbol{T}_{SegementedVertebraLx\_ISOCENTER}$ using (*134*).

Once the registration is achieved, the obtained registration pose for the segmented vertebra would be ${}^{DRR}\boldsymbol{T}_{SegementedVertebraLx}$, which can be transformed to {DICOM} using the following relation:

$$ {}^{DRR}\boldsymbol{T}_{DICOM_{VertebraLx}} = {}^{DRR}\boldsymbol{T}_{SegementedVertebraLx} \cdot \left({}^{DICOM}\boldsymbol{T}_{SegementedVertebraLx}\right)^{-1} \qquad (135) $$

The result of equation *(135)* is, in fact, ${}^{DRR}\boldsymbol{T}_{DICOM}$, but for clarity, the resulting transformation is named ${}^{DRR}\boldsymbol{T}_{DICOM_{VertebraX}}$, which indicates the local registration, ${}^{DRR}\boldsymbol{T}_{DICOM}$, of the vertebra X.

Joining Local Registrations to Build a Quasi-Global Registration

Screws and implants are planned in the pre-operative data, i.e., CT-scan with DICOM format. As explained in section 5.2.7, the planned data has its coordinate in {DICOM}, meaning that the screw pose is given by a transformation matrix of the type ${}^{DICOM}\boldsymbol{T}_{IMPLANT}$. From the DICOM segmentation, each vertebra boundaries are known by the segmentation cube, which is also described in {DICOM}.

When a planned implant lies within the boundaries of a segmentation cube, the local registration of that segment is assigned to that specific implant. That means a ${}^{DICOM}\boldsymbol{T}_{IMPLANT_X}$ is assigned a ${}^{DRR}\boldsymbol{T}_{DICOM_{VertebraLx}}$ based on the vertebra, where it is planned. In other words, the global registration is approximated as a piecewise-defined function in the volume space, {DICOM}. Therefore, this approach is called a quasi-global 2D/3D registration.

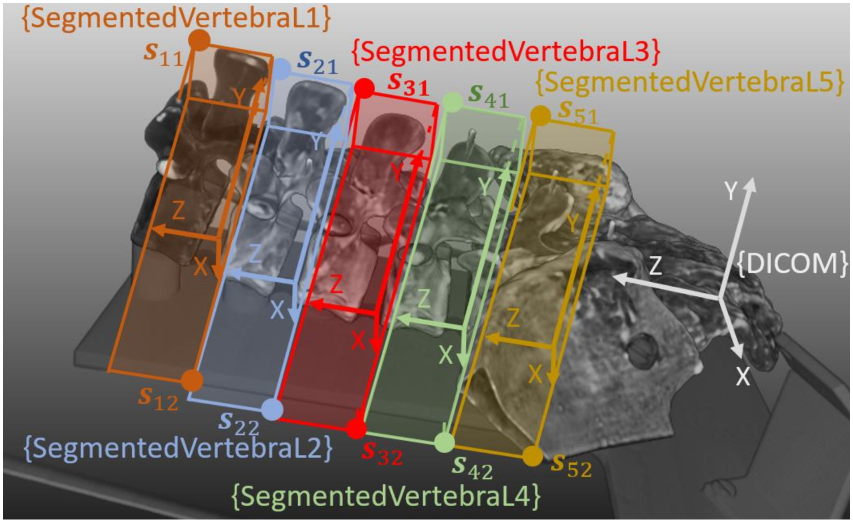

Figure 5-54. 3D volume rendered in a MeVisLab examiner viewer

Formally, the global registration is defined as follows:

$$ {}^{DRR}\boldsymbol{T}_{DICOM} = \begin{cases} {}^{DRR}\boldsymbol{T}_{DICOM_{VertebraL1}} & if \quad x_{11} \le x \le x_{12} \cup y_{11} \le y \le y_{12} \cup z_{11} \le z \le z_{12} \\ \vdots & \qquad \vdots \\ {}^{DRR}\boldsymbol{T}_{DICOM_{VertebraLx}} & if \quad x_{x1} \le x \le x_{x2} \cup y_{x1} \le y \le y_{x2} \cup z_{x1} \le z \le z_{x2} \end{cases} \qquad (136) $$

It can be noticed from (*136*) that outside the boundaries, the registration is undefined. This could be solved by assigning the closest transformation to the required point in space. However, as this application is made for spine surgery, an implant planned out of boundaries, i.e., not in a vertebra, makes no sense. From an application point of view, the non-definition of the registration can be seen as a feature for detecting wrongly planned implants.

5.8.3. Discussion and Conclusions

The results of 500 registrations show that the accuracy of planned screws inside the ROI is higher than the accuracy of planned screws outside the ROI. A local 2D/3D registration is the registration executed over a defined ROI. The area squared by the ROI in the X-ray image gives the most accurate results on the corresponding part of the 3D volume.

The DICOM segmentation is created by trimming the entire 3D volume into a manually defined cube containing a vertebra. There are two things to consider from the segmentation: the cube vertices and the transformation from the DICOM volume to the defined vertebra reference frames, ${}^{DICOM}\boldsymbol{T}_{SegementedVertebra}$.

The quasi-global 2D/3D registration is approximated by the union of multiple local 2D/3D registrations. This union is expressed as a piecewise function that depends on the DICOM coordinates. As the screws and implants are planned in {DICOM}, the assignment of a local 2D/3D registration is dependent on the position of the implants within the DICOM coordinates.

6. Accuracy of the Selected 2D/3D Image Registration Approach

During chapter 5, the implementation of the 2D/3D registration is explained component by component. A set of experiments using a well-registered device are carried out to determine the C-arm characterization accuracy. In section 5.5, a set of registrations using a lumbar spine phantom is executed to determine the most suitable image similarity measurement and optimization algorithm. In section 5.8, another set of registrations are made using the spine phantom to test the local registration hypothesis based on the ROI selection. These previously executed registration tests have a common particularity; the results rely only on the error of the registration due to the translation mismatch, equation *(104)*, as explained in section 5.2.7.

The use of the error of the registration due to the angle mismatch, equation *(106)* in chapter 5.2.7, requires additional care in the experimental setup. Recalling from chapter 5.2.7, the required measures to compute the registration error are the transformations ${}^{TRB}\boldsymbol{T}_{Landmark}$ and ${}^{DICOM}\boldsymbol{T}_{Landmark}$. These transformations are obtained from a corresponding landmark in the phantom and the CT-scan. To measure the angle error, the landmark, which is pointed by a planned implant, has to be planned with the exact orientation as the measured implant in the phantom. These former conditions are hardly achievable with the experimental setup presented so far.

In this chapter, a new setup is explained for measuring the 2D/3D registration error, such as both errors of the registration due to the translation and orientation mismatch can be accurately computed. Also, the experiment execution is explained, and the results are analyzed. After that, the accuracy value of the implemented 2D/3D registration is approximated based on the normal distribution of the results. This chapter starts with a data analysis tool to assess normally distributed data sets.

6.1. Assessing Normal Distribution in Data Sets

Two key statistical concepts in data analysis that can determine the normal distribution of a data set are the skewness and the kurtosis values.

Skewness gives an estimation of the data symmetry. Positive skewness indicates values gather to the left and spread with a long tail to the right, and negative skewness indicates values gather on the right with a tail to the left. See Figure 6-1.

The kurtosis value has been defined as the measure of the data distribution peakedness with a positive kurtosis value indicating a central-peaked distribution with long thin tails. A negative kurtosis value indicates a flat distribution with many values on the extremes. See Figure 6-2. However, a publication in 2014 removes the peakedness interpretation and transforms the definition only to the tail extremity, i.e., either measuring existing outliers or expressing the tendency to produce outliers [180]. In a normal distribution, skewness and kurtosis have a value of zero [181].

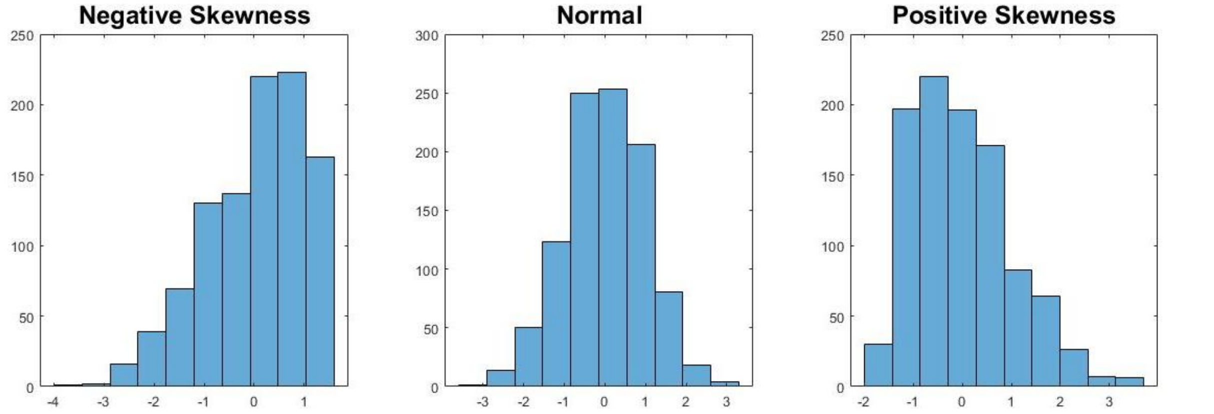

Figure 6-1. Data distributions with different skewness: Negative skewness on the left, zero skewness on the center, and positive on the right.

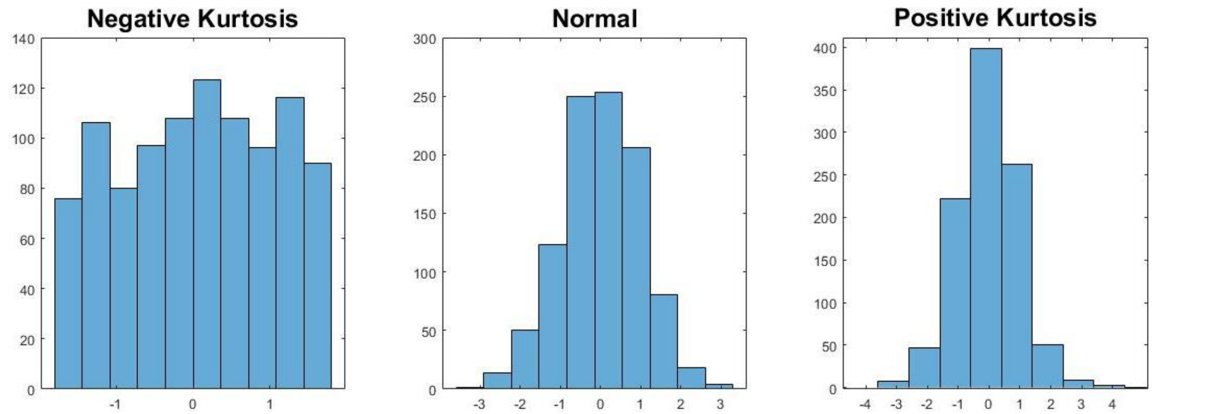

Figure 6-2. Data distributions with different kurtosis values: Negative kurtosis value on the left, kurtosis equals zero on the center, and positive kurtosis value on the right

The normality assessment is computed based on a z-test, which calculates z-scores as the quotients of the skewness and kurtosis values with their respective standard errors. The z-scores equations are written in *(137)* [182].The equations for calculating skewness, kurtosis, and standard errors are not explicitly written in this work, but they can be found in [183].

$$z_{skewness} = \frac{skewness}{S_{ErrorSkewness}}$$
$$z_{kurtosis} = \frac{kurtosis}{S_{ErrorKurtosis}} \quad (137)$$

The assumption of normality is divided into three different criteria depending on the number of samples. Each of those divisions compares the z-scores with a different threshold, gathered in Table 5-14 [182].

Table 6-1. Determination of normally distributed data based on z-scores

Sample size	Strategy	Criteria to assess data normally distributed
<50	Use equation (*137*)	$\lvert z_{skewness} \rvert < 1.96 \wedge \lvert z_{kurtosis} \rvert < 1.96$
50 to 300	Use equation (*137*)	$\lvert z_{skewness} \rvert < 3.29 \wedge \lvert z_{kurtosis} \rvert < 3.29$
+300	Use $skewness$ and $kurtosis$	$\lvert skewness \rvert < 2 \wedge \lvert kurtosis \rvert < 3$

6.2. Experiment Description for Measuring Position and Orientation Accuracy

The necessary conditions to calculate the 2D/3D registration accuracy are described in detail in section 5.2.7. It is required to have a patient, represented by a spine phantom in this case, and its DICOM volume. An implant or screw is located in both the phantom and DICOM volume, so the 2D/3D registration result is used as a bridge to find inaccuracies between the planned and executed implant pose. Measuring the implant pose in the DICOM, ${}^{DICOM}\boldsymbol{T}_{Landmark}$, is done using the planning software, while the implant pose in the phantom, ${}^{DRB}\boldsymbol{T}_{Landmark}$, is obtained using the navigation system, an RB installed in the phantom and a pointer. See section 5.5.2.

The accuracy measurement is split in error due to the translation mismatch, equation *(104)*, and the error due to the angle mismatch, equation *(106)*. The implant position must be aimed at the same landmark in the DICOM and phantom to measure the translation mismatch. Similarly, determining the angle mismatch requires the same implant orientation to the same landmark in the DICOM and phantom. A device is used to ensure the implant positioning in the phantom in the experiment performed in section 5.5.2. The exact position can be replicated without any difficulty in the planning software.

However, corresponding the phantom and planning software orientations using the section 5.5.2 setup is prone to inaccuracies. The interaction between the phantom and pointer contains an angle backslash. It induces an inaccuracy of a couple of degrees between the screw in the planning software and the phantom. As a result, the setup in section 5.5.2 cannot accurately calculate the measurement of the error due to the angle mismatch. The graphical description of the orientation inaccuracies is shown in Figure 6-3.

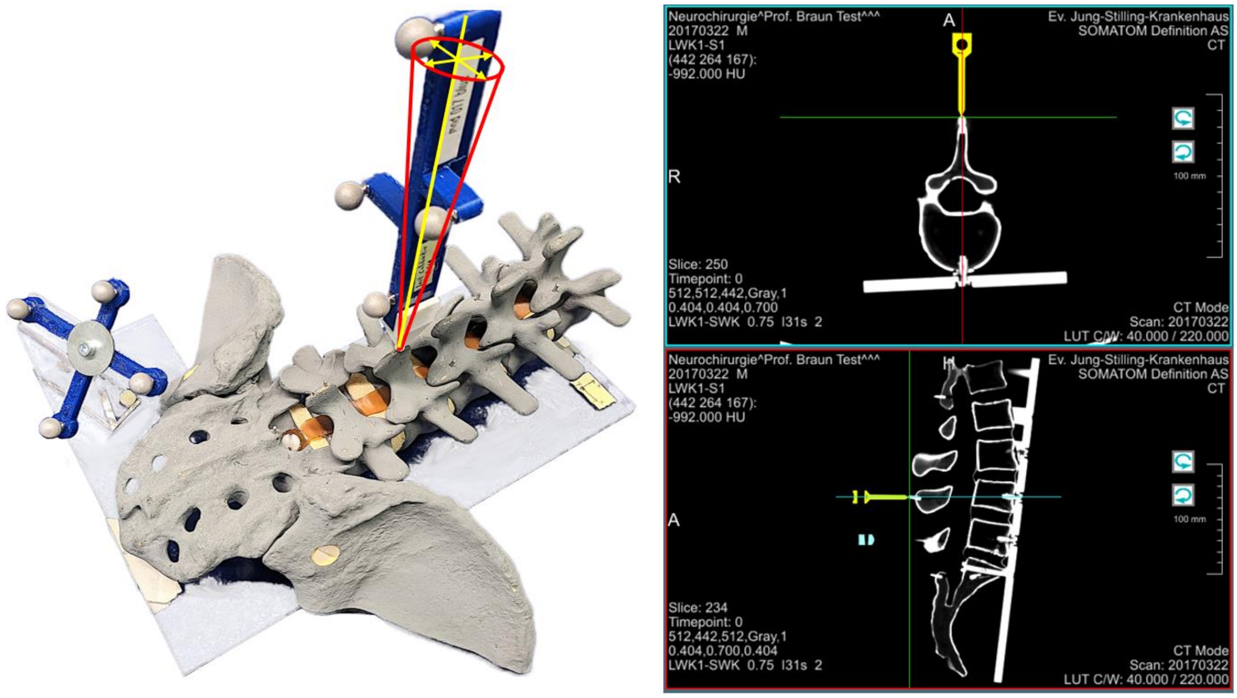

Figure 6-3. Screw angle inaccuracies between the planning software and phantom due to current setup

The implant in the planning software can be seen on the right side of Figure 6-3, and on the left side, it can be seen the real-scenario setup. The main axis of the pointer is depicted in yellow, and around the main axis, a red cone illustrates the orientation uncertainty due to the setup.

New Setup for the 2D/3D Registration Error Measurement

A single vertebra model is used to avoid the source of angle inaccuracies. The new phantom, being a single vertebra, produces a local registration result without introducing manual segmentation in the DICOM. Therefore, the result obtained from the new setup gives a local 2D/3D registration, which, as explained in section 5.8, are the type of results obtained from the process implemented in this work.

This single vertebra model is a 3D-printed device containing two channels of 5.1mm in diameter with an incidence of 15 degrees. The new phantom is depicted in Figure 6-4.

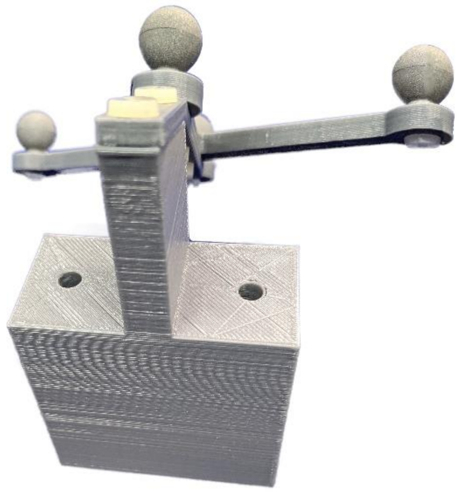

Figure 6-4. Phantom for measuring the 2D/3D registration accuracy

The two channels ensure that the orientation and position of the pointer in the real-scenario are fixed and can be duplicated accurately in the planning software. In each channel, an implant is considered, such as there are two error measurements per registration, i.e., ${}^{DRB}\boldsymbol{T}_{LandmarkLeft}$ and ${}^{DRB}\boldsymbol{T}_{LandmarkRight}$.

As the pointer tip has a diameter of 5mm and 35mm length, it enters in the phantom without backslash and remains stable during the measurements. The setup for finding ${}^{DRB}\boldsymbol{T}_{LandmarkLeft}$ and ${}^{DRB}\boldsymbol{T}_{LandmarkRight}$ can be seen in Figure 6-5, where {PRBLeft} is equivalent to {LandmarkLeft}, and {PRBRight} is equivalent to {LandmarkRight}.

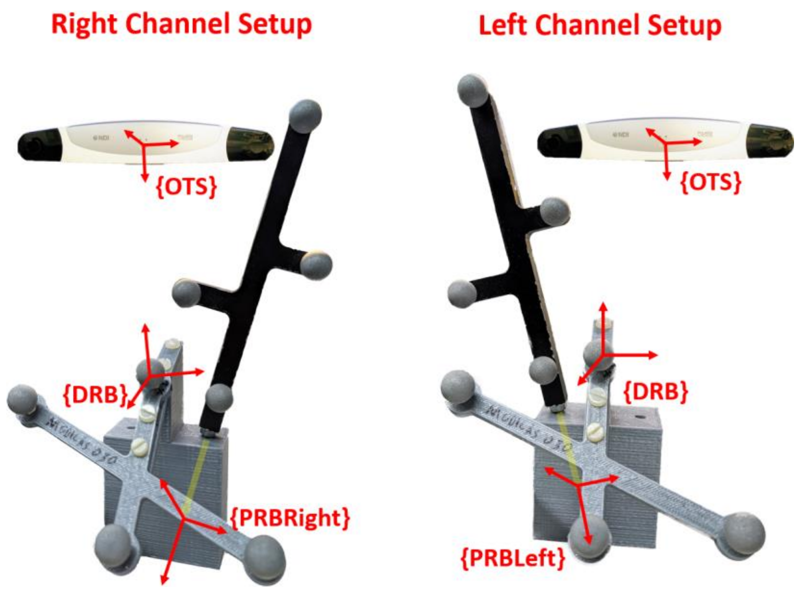

Figure 6-5. ${}^{DRB}\boldsymbol{T}_{LandmarkRight}$ and ${}^{DRB}\boldsymbol{T}_{LandmarkLeft}$ explanation

For calculating the error, it is necessary also to find ${}^{DICOM}\boldsymbol{T}_{LandmarkLeft}$ and ${}^{DICOM}\boldsymbol{T}_{LandmarkRight}$. The new phantom is developed in a computer-aided designed (CAD) software in a typical format for 3D-printers, Stereo Lithography (STL) file, but the planning software requires a DICOM format. The conversion from STL to DICOM is done using an own implementation. The conversion details can be found in Appendix A.8. Once the DICOM of the new phantom is obtained, the planning is done using a 35mm implant, the same size as the pointer length. The planning views are seen in Figure 6-6, where {ImplantLeft} is equivalent to {LandmarkLeft}, and {ImplantRight} is equivalent to {LandmarkRight}.

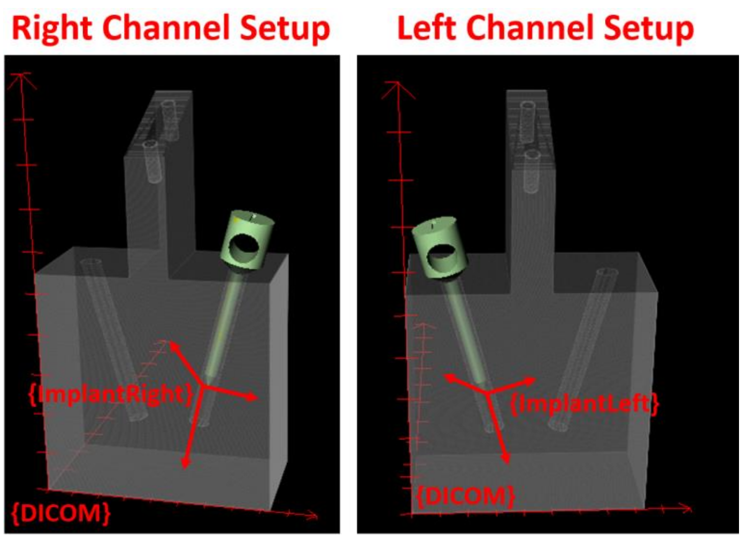

Figure 6-6. ${}^{DICOM}\boldsymbol{T}_{LandmarkRight}$ and ${}^{DICOM}\boldsymbol{T}_{LandmarkLeft}$ explanation

6.3. Experiment Execution for Measuring the Position and Orientation Accuracy

To measure the 2D/3D registration accuracy, ten AP and ten LAT X-rays of the new phantom device are taken. In Figure 6-7, it can be seen a typical setup for capturing AP and LAT X-ray images of the new model using the C-arm.

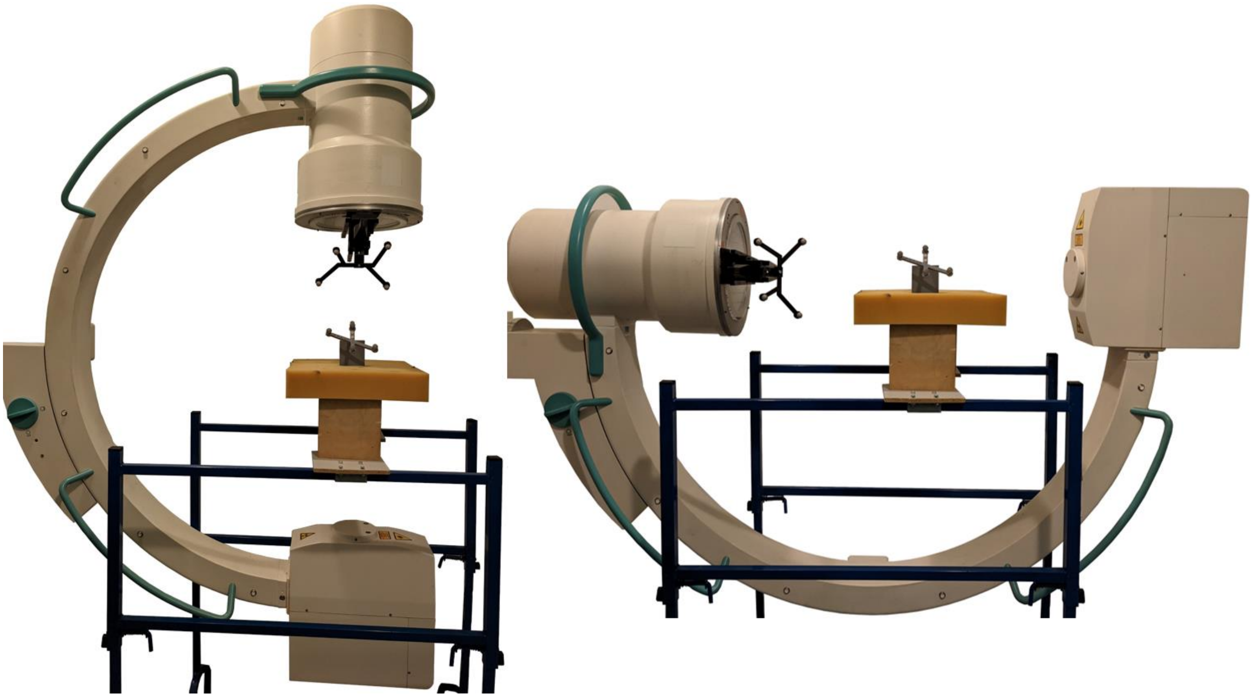

Figure 6-7. AP setup of the new phantom on the left side, and LAT setup on the right side

The resulted AP and LAT X-ray images of the setup in Figure 6-7 can be seen in Figure 6-8.

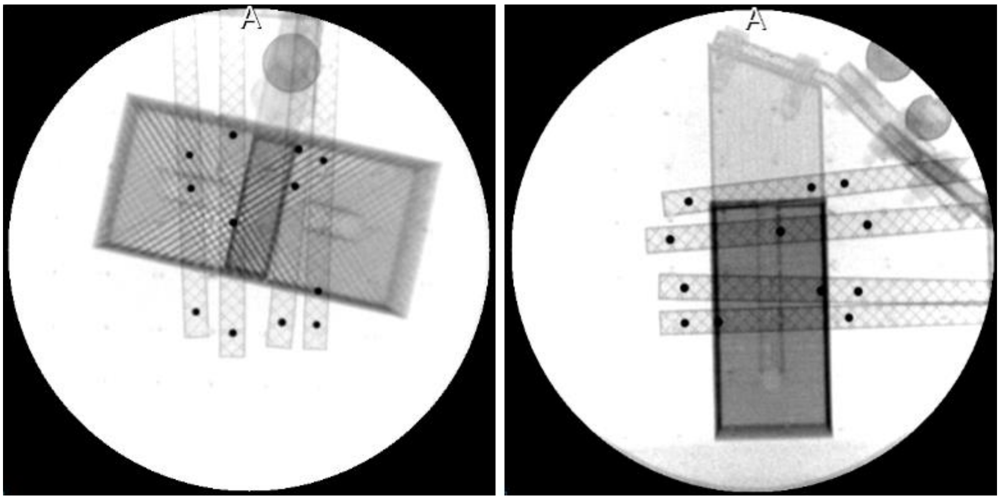

Figure 6-8. X-ray image with AP projection of the new phantom on the left side, and X-ray image with LAT projection on the right side

The pair combinations of the ten AP images with the ten LAT images result in 100 pairs of AP-LAT images. Each of these combinations has two implants defined in the planning software, giving two error measurements per registration. That means the experiment results contain 200 registration errors, used to find the 2D/3D registration accuracy. It is worth mentioning that equations *(104)* and *(106)* are used in this case; using the results of the 200 registrations, the RMS errors due to the translation and angle mismatch are computed. The angle mismatch in *(106)* interprets the angle inaccuracy as a displacement in the entry point of the implant; therefore, the units of the angle mismatch are millimeters.

6.4. Result Analysis of the 2D/3D Registration Accuracy

The RMS errors of the registration due to the translation and angle mismatch are plotted as two histograms, shown in Figure 6-9 and Figure 6-10.

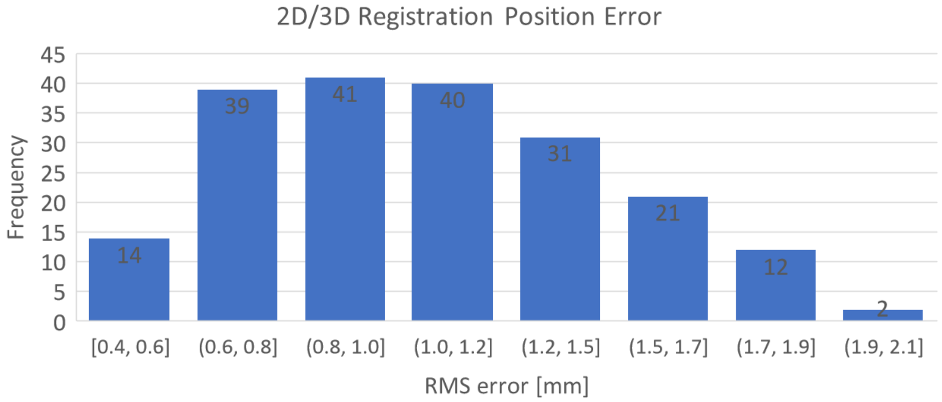

Figure 6-9. Histogram of the 2D/3D registration RMS error due to the translation mismatch

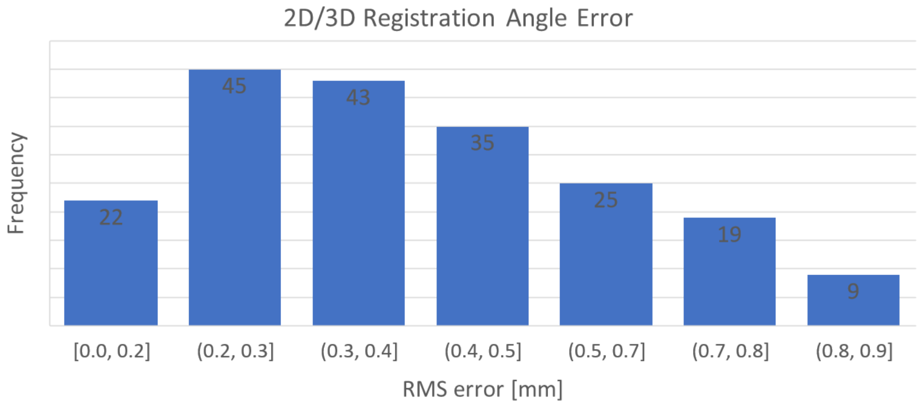

Figure 6-10. Histogram of the 2D/3D registration RMS error due to the angle mismatch expressed in millimeters

Additionally, the total registration error of the 2D/3D registration is computed using the previous results and the equation *(108)*. The RMS of the total registration error is plotted as a histogram in Figure 6-11.

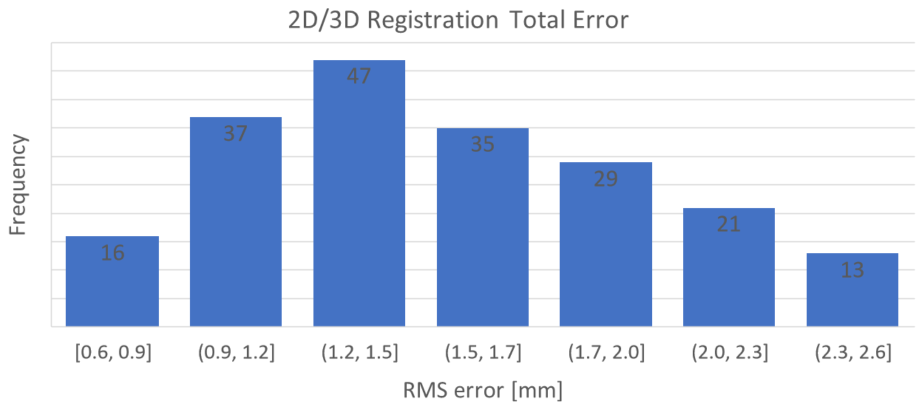

Figure 6-11. Histogram of the total 2D/3D registration RMS error

The previous histograms show a standard distribution shape, but the values gather a bit to the left and spread with a tail to the right. It has much sense that the data displays a gaussian distribution, based on the central limit theorem [167], as there are multiple sources of error associated with the 2D/3D registration process, for instance, quantization errors in the X-ray and DRR images, C-arm frames characterization errors, errors in the RBs poses reported by the navigation system, and others. As the histograms have a similar shape to the normal distribution, the data could be already considered normally distributed; following Wheeler's recommendation: "at times, it can be best to simply inspect histograms to determine if the data is normally distributed" [183]. In addition to the arguments mentioned above, an analytical approach is also used to assess the normality of the results by means of the skewness and kurtosis values, as explained in section 6.1.

In this case, the sample size is 200. Based on Table 5-14, the assumption of normality is valid when the absolute value of both z-scores, calculated with equation *(93)*, are smaller than 3.29. The data analysis of the RMS errors due to the translation, angle, and total mismatch are presented in Table 6-2.

Table 6-2. Data analysis of the RMS errors due to translation, angle, and total mismatch

	Translation RMS Error	Angle RMS Error	Total RMS Error
Mean [mm]	1.09	0.41	1.49
Standard Deviation [mm]	0.37	0.22	0.48
Kurtosis	-0.77	-0.83	-0.68
Standard Error Kurtosis	0.35	0.35	0.35
Skewness	0.24	0.41	0.37
Standard Error Skewness	0.17	0.17	0.17
$z_{skewness}$	1.36	2.37	2.14
$z_{kurtosis}$	2.21	2.37	1.95

The z-scores, $z_{skewness}$ and $z_{kurtosis}$, for the translation error are 1.36 and 2.21, for the angle error 2.37 and 2.37, and 2.14 and 1.95 for the total error, respectively. All of those z-scores satisfied the condition in Table 6-1. It can be concluded that a normal distribution represents the RMS errors results of the 2D/3D registration tests. That means the accuracy of the 2D/3D registration within a confidence interval of 68.27% can be calculated as the mean value plus-minus one standard deviation. Bearing that in mind, the 2D/3D registration accuracy considering only the translation match is 1.09 ± 0.37mm. The accuracy of the 2D/3D registration only considering the angle mismatch is 0.41 ± 0.22mm, and the total accuracy of the 2D/3D registration is 1.49 ± 0.48mm.

6.5. Discussion

A new vertebra phantom is developed, containing two channels of the same diameter as the pointer to ensure the orientation of the pointer in real-scenario and guidance in the planning software. With the previous constraints guaranteed, the angle error is measured safely.

The total 2D/3D registration error is measured using ten AP and ten LAT X-ray images. The combination of AP with LAT image creates 100 AP-LAT pairs. A registration is executed per each of the 100 combinations, and for each registration, two implants are planned. As each implant gives an error measurement, 200 measurements are used to measure the 2D/3D registration accuracy.

The results show that the 2D/3D registration error due to the translation mismatch, 2D/3D registration error due to the angle mismatch, and the total 2D/3D registration accuracy follow a normal distribution. It is found that the implemented 2D/3D registration process has an accuracy of 1.49 ± 0.48mm within a confidence interval of 68.27%.

7. Improvements to the 2D/3D Registration Using Deep-Learning

In the following sections, a novel automatization of the X-ray image undistortion in conventional C-arms, section 7.1, and the initial pose selection, section 7.2, are developed using a deep-learning approach.

The implemented deep-learning models are described in a result-wise style instead of process-based. That means the final topology for the specifics approaches are explained, and their results are shown. The decision-making process that brings the implementations to their final shape is omitted.

This work does not focus on low-level details of deep-learning implementations but shows the possibilities and advantages of incorporating deep-learning in the 2D/3D registration field.

7.1. Fiducial Detection in X-Ray Images for Undistortion of Conventional C-Arm Images Using Convolutional Neural Networks (CNN)

As seen in section 5.3, X-ray images from conventional panel C-arms contain distortions mostly due to the earth's magnetic field effects on the C-arm image intensifier [28]. The undistortion of X-ray images relies on the detection of the centers of the fiducials in the radiograph. During section 5.3.3, it is considered that the fiducial detection is done manually; therefore, the automation of the fiducial detection is a logical step to facilitate the 2D/3D registration procedure. In this section, the fiducial detection is tackled using a classical image processing approach, and later, compared with a novel implementation based on deep-learning [184]. It is seen that the deep-learning approach gives more beneficial results for a real-case scenario than the classical image processing implementation.

7.1.1. Structure of the Fiducial Detection Using Image Processing

The classical image processing approach for detecting fiducials is made as a combination of image processing algorithms found in the literature. It starts by applying a median filter to the original image. This filtered image is subtracted from the original image to make a basic background removal [162]. Later, the fiducial detection is carried out on the image without background using the circle Hough Transform (CHT), a commonly feature-extraction algorithm used in image processing [185]. The CHT uses the Canny edge detector to identify the pixels forming edges. These pixels are organized in sets. Each set contains the pixels that belong to a circumference of radius R. The pixels in each set are used to find the center of the circumference.

As the used fiducials have a fixed size, the circumference radius for the CHT algorithm is delimited. After a circumference is detected, the center of the fiducials can be computed. However, a problem arises with the Canny edge detector, which uses a threshold to discern the gray levels between objects and the background [186]. The threshold variation impacts the fiducial detection; thus, the Canny threshold is kept as a user parameter. Figure 7-1 shows the detection differences by using the same image with different Canny thresholds. It can be noticed that on the left side, 49 fiducials are detected, compared with 57 on the right.

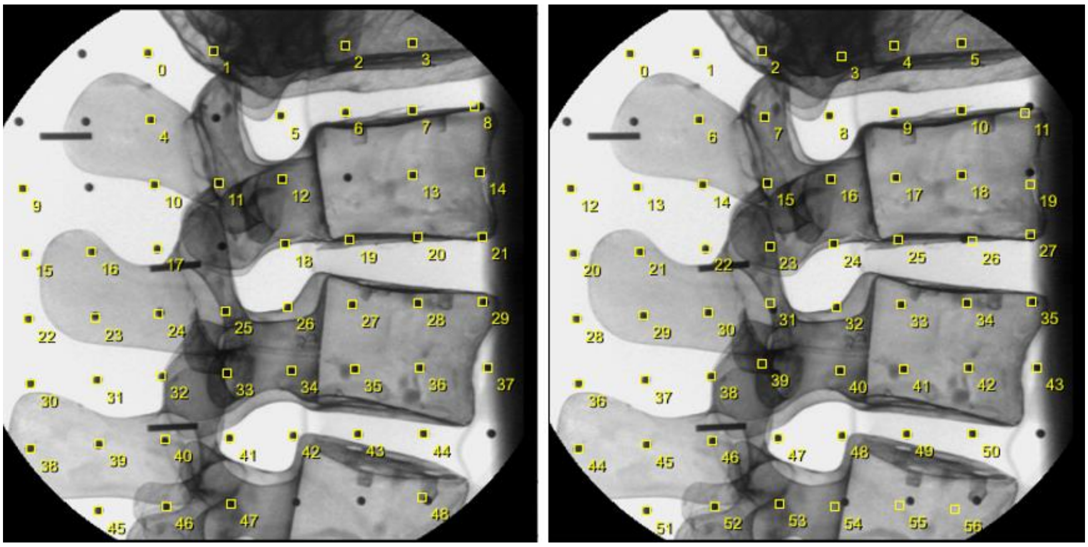

Figure 7-1. Fiducial detection algorithm based on Hugh transformation. On the left, an used a Canny threshold of 20 and 100 on the right

Different variations among X-ray images, such as changes in patient anatomies, impact the contrast of the images; thus, finding a proper Canny threshold is not always possible. Indeed, the threshold must be adapted in each image.

To avoid parameter tuning in the undistortion process and improve the detection rate, a CNN is implemented. The classical image processing approach is considered as the ground truth for the detection accuracy and rate. It also supports the dataset labeling process, the earliest stage of CNN development.

7.1.2. Structure of the Fiducial Detection Using Deep-Learning

For implementing the deep-learning approach, a particularity of the image distortion approach, section 5.3, is used in favor. Regardless of the image distortion, it is possible to define squared zones centered on the ideal position of the fiducials, so that the fiducials fall inside those zones. A squared zone that includes a fiducial is determined to be about 27x27px. Every image is divided into 60 sub-images, with every sub-image containing one fiducial, but each zone size is taken as 32x32px. It increases the security factor, i.e., a sub-image always contains a fiducial, and facilitates the max-pooling operations on the CNN, i.e., dividing the sub-image size by two. With this approach, the implemented CNN focuses on detecting 3mm diameter fiducials within a grayscale image of 32 by 32 pixels. In a previous stage, the offset of each sub-image with respect to the image is stored, such as the inferred position is added, giving the fiducial prediction in the reference frame of the original image. This previous process is repeated in every fiducial, i.e., sixty times per image.

Data collection and labelling

Using the Ziehm Vario 3D C-arm, eight scans with the C-arm are made. Each scan is composed of 140 images, giving 1120 images in total. Each scan covers the entire C-arm range of movement, 130 degrees. Afterward, every X-ray is split into 60 sub-images. Each of these sub-images is labeled with the X and Y coordinates of the fiducial center position. The dataset used to implement the convolutional neural network (CNN) is formed by 67200 sub-images. The image processing method, implemented in section 7.1.1 for fiducial detection, supports the dataset labeling. The dataset is split into 60% for training, 20% for validating, and 20% for testing. During the CNN training and later comparison with the image processing method, the prediction error is calculated as the distance between the predicted and the actual fiducial center. The error is normalized using the maximum possible error, the diagonal of the 32x32 pixel sub-image, i.e., $32\sqrt{2}$ pixels. The prediction accuracy is calculated as the unit minus the normalized error.

7.1.3. Development of the CNN Layout

The selected topology consists of four convolutional layers. A max-polling layer follows each convolution except in the last convolution layer. The last convolutional layer output is connected to the input of a fully connected neural network (FCNN). This FCNN consists of one input, two hidden, and one output layer. The CNN layout can be seen in Figure 7-2, and it is described in detail in Table 7-1.

Table 7-1. CNN Topology

Layer	Type	(Out)-Size	# Channels	Kernel Size	Stride	Padding	Activation
Input	Image	32x32	1	-	-	-	-
2	Convolution	32x32	18	5x5	1	2	ReLU
3	Max Pooling	16x16	18	2x2	1	2	ReLU
4	Convolution	16x16	32	5x5	1	2	ReLU
5	Max Pooling	8x8	32	2x2	1	2	ReLU
6	Convolution	8x8	64	5x5	1	2	ReLU
7	Max Pooling	4x4	64	2x2	1	2	ReLU
8	Convolution	4x4	128	5x5	1	2	ReLU
9	FCNN	1024	-	-	-	-	ReLU
10	FCNN	120	-	-	-	-	ReLU
Output	FCNN	2	-	-	-	-	Linear

This topology is similar to a classical CNN topology called LeNet-5 [187], but instead of Tanh, the activation functions are replaced with ReLU. In contrast with LeNet-5, the implemented net propagates through all of the channels (features maps) from layer to layer, increases the number of feature maps used per layer, uses a bigger FCNN at the end of the network, and the network output uses a linear activation function instead of a classifier as done by Toshev and Szegedy [188]. The idea behind using a regression-based model is to predict the fiducial center coordinate. These changes are developed following the new improvements done in recent years to deep-learning approaches [189]. The CNN model is trained using as cost function the mean squared error (squared L2-norm) and the stochastic gradient descent with a momentum of 0.9 as optimizer. The selected cost function and optimizer are typical settings for training deep-learning models. Through trial and error, the learning rate for this problem is found to be 0.3. The training is run through 100 epochs.

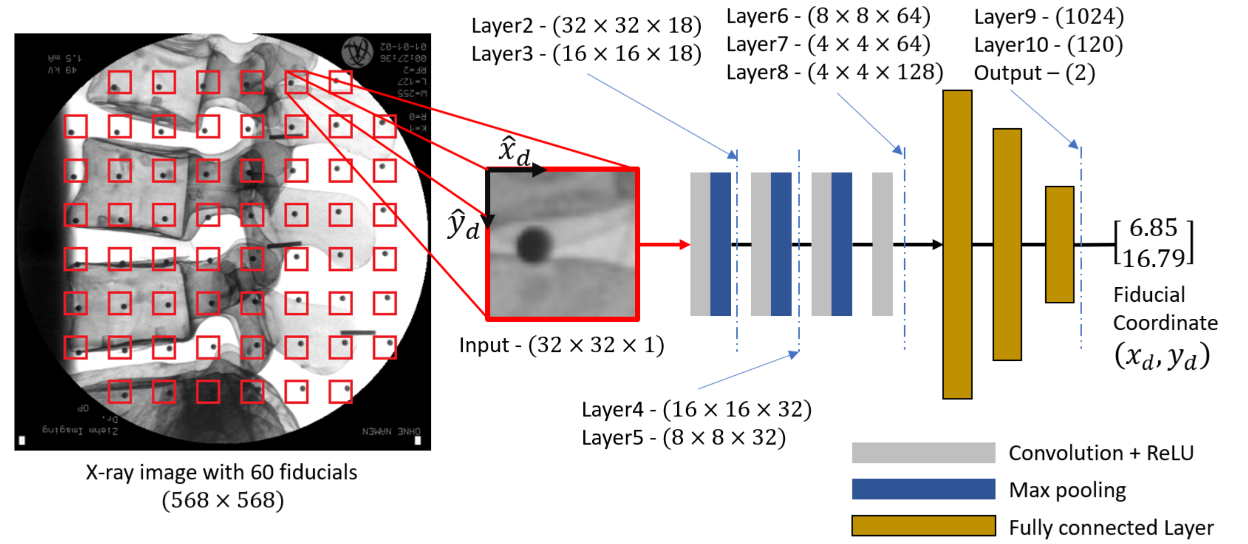

Figure 7-2. CNN topology with input and outputs

The training is carried out using the training dataset, and the prediction accuracy is calculated from the training and validation dataset. The training dataset accuracy keeps increasing, 99.55%, and the training loss decreasing, 0.000197, until the reach of the 100th epoch. On the other hand, the validation dataset reaches a plateau in epoch 60 when its accuracy, 99.234%, and the validation loss, 0.00124, remain steady, as Figure 7-3 depicts. Consequently, an early stop mechanism is used in epoch 60, where the CNN model shows a good fit for the problem.

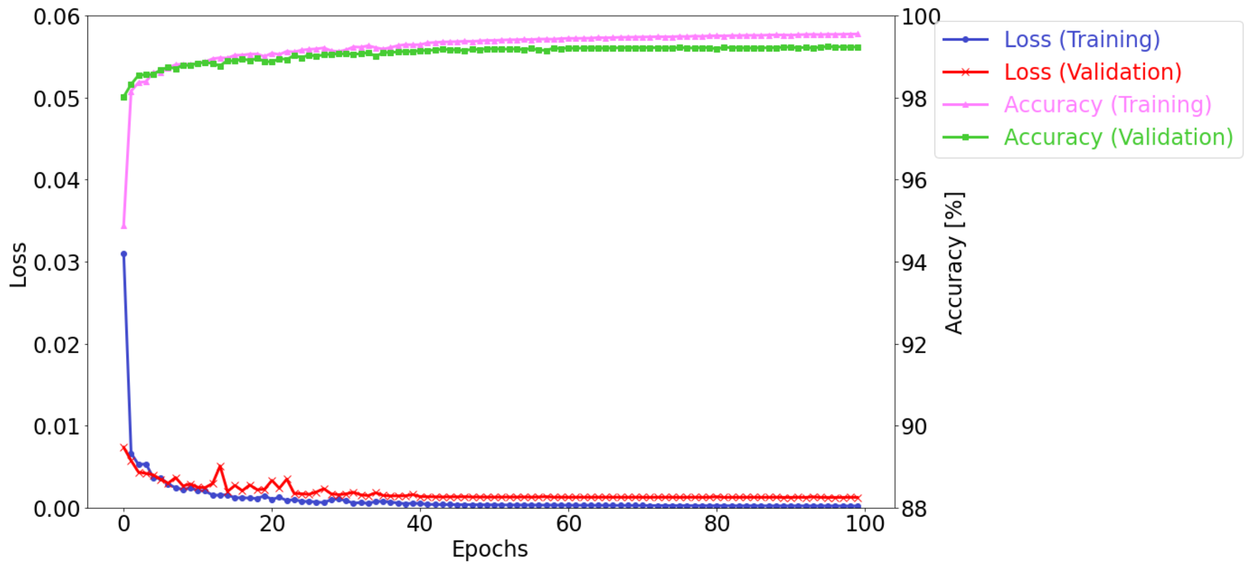

Figure 7-3. Training plot of the CNN approach for the fiducial detection

7.1.4. Fiducial Detection: CNN vs. Image Processing Approach

The following results are executed over the testing dataset, i.e., 224 remaining images (20% of the 1120 initial X-ray images.) The CNN predictions are compared against the image processing method based on the CHT. The results are shown in Table 7-2. It is found that the detected fiducials using the traditional method, in which the Canny threshold can be changed, gives a prediction accuracy of 99.13% when the threshold is set to 100. Nevertheless, it gives a detection rate of 89.03%, equivalent to detect 53 out of 60 fiducials. The detection rate increases to 97.71% when the Canny threshold is set to 20, but the prediction accuracy decreases to 97.01%.

On the other hand, the CNN gives a prediction accuracy of 99.24%, which surpasses the average accuracy of the conventional method using a Canny threshold in 100, and it increases the fiducial detection rate to 100% compared to 97.71% when using the Canny threshold set to 20.

Table 7-2. Accuracy results of tested methods for fiducial detection

Method	Canny Threshold	Prediction Error [%]	Detection Rate [%]
Image processing	20	97.01	97.71
	50	98.48	91.9
	80	98.98	89.89
	100	99.13	89.03
CNN	-	99.24	100

As the CNN does not require any user-defined parameters after it is already trained, it can be said that the CNN successfully automates the fiducial detection by its intrinsical working principle of inferencing a result only based on the current input. With that said, asides from the low prediction error and the high detection rate, the CNN implementation minimizes the user intervention to the minimum extent. It can be concluded that the CNN achieves the automation of the X-ray image undistortion. The runtime for loading the CNN in the memory is less than one second. The inference of one sub-image takes around 30ms, and the fiducials prediction over the entire image (60 sub-images) is computed in less than two seconds. The automatic fiducial detection requires three seconds in total. The process of having an X-ray image, detecting fiducials, undistorting, and inpainting the image can be seen in Figure 7-4.

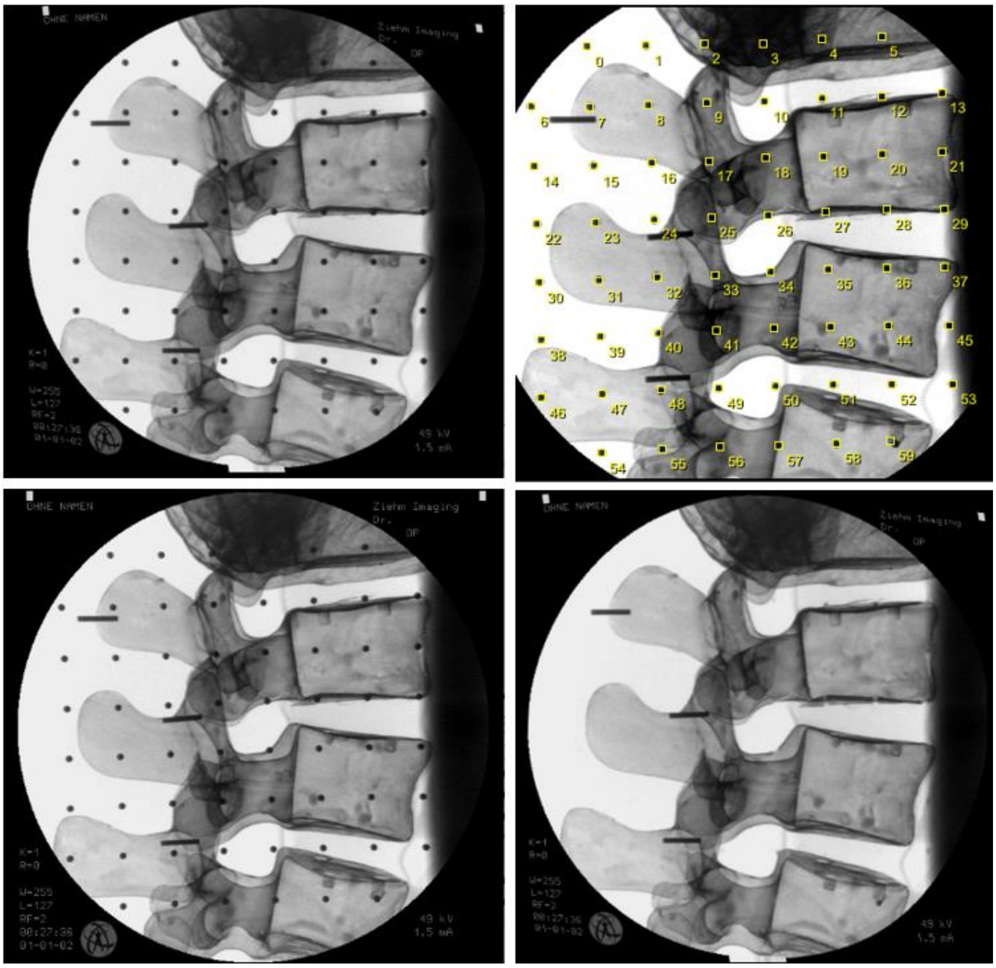

Figure 7-4. Original image with fiducials on the upper-left. Result of inference with CNN on the upper-right. Result of the undistortion on the bottom-left. On the bottom-right, undistortion after inpainting

7.1.5. Discussion

A conventional image processing algorithm based on the circle Hough transformation is implemented to automate the fiducial detection in X-ray images. It gives a prediction accuracy of 99.13% in the best case, but the detection rate is not good enough to be used as an automatic process to remove distortions in X-ray images as it fails to detect seven fiducials on average.

The conventional image processing algorithm for detecting fiducials is used to establish a ground truth and support the data labeling process to develop a deep-learning approach.

The implementation of the CNN shows that the prediction accuracy is higher than using the conventional image processing method, 99.13% and 98.24%, respectively. The image preparation for the deep-leaning approach intrinsically increases the detection rate to 100%. These results fill the expectations to create an automatic unit for undistorting conventional C-arm images.

7.2. Automatic Initial Pose Generator for the 2D/3D Registration Using CNN

It is seen in section 5.8 that the initial pose for the 2D/3D registration is selected using a graphical method. The user must select an initial pose manually, which renders a DRR image from the CT-scan that is registered. The initial pose is considered suitable if the rendered DRR image looks similar to the X-ray image to be registered. The manual selection of the initial pose presents two significant disadvantages. It is quite probable to end up selecting an initial pose that is not close enough to the actual registration pose, +/- 5 degrees and +/- 10mm in each axis [190]. Also, the manual pose selection is a time-demanding process.

Those disadvantages present a risk to the 2D/3D registration accuracy and repeatability. One approach to deal with these drawbacks is to automate the pose generator. Some attempts in the literature to automate the initial pose of the 2D/3D registration have been tested using a set of random initial poses [81] and pre-computing sets of 2D templates of a wide range of 3D poses [191]. Although these approaches automate the initial pose selection, they do not guarantee the nearness of the initial pose to the actual registration pose; therefore, the registration accuracy is not ensured. In some studies, deep-learning approaches have been introduced to replace the optimizer and cost function [147], but there is no work in the literature suggesting an initial pose generator based on deep-learning up to the elaboration date of this thesis.

The implemented 2D/3D registration process shows promising results as long as the initial pose is adequate. For this reason, a deep-learning approach is presented in this section, such as a CNN infers the initial pose, and then it is given to the 2D/3D registration process. The mix of the initial pose generator and the current registration process forms a fully automatic 2D/3D local registration.

7.2.1. Structure of the Initial Pose Generator Using Deep-Learning

For manually selecting an initial pose for the 2D/3D registration, there are two inputs required, an X-ray image and a DICOM volume. The same two inputs are used for the CNN, but the DICOM is not inputted as a 3D object. Instead, the DICOM is converted into two radiographs, one AP and one LAT projection, using the DRR module. That means the CNN inputs are three 2D images, one AP and one LAT DRR projection of the DICOM, and the X-ray image to be registered. Some CNNs for tuberculosis diagnosis use CT-scans as input, meaning the CNN processes the entire 3D volume [192]; however, implementing a 3D CNN requires more weights to be optimized than a 2D CNN.

In this case, most acquired data belongs to the X-ray image side, meaning most of the changing data is available on the 2D CNN side. In that way, the training of a 3D CNN topology would have fewer chances to be successful. Although some data augmentation techniques increase the 3D data, these pitfalls are avoided by transforming the implementation into a CNN with three 2D inputs.

The CNN output is the DRR pose that makes the DRR image similar to the inputted X-ray. Following the notation introduced in equation *(47)*, the CNN output is the six DoF vector $\vec{\boldsymbol{p}}$. It is worth noting that the CNN is trained to predict only X-ray AP poses, as the initial LAT pose is analytically found using equation *(97)*.

Acquisition of the Dataset for Developing the CNN

It is estimated that capturing and labeling one single X-ray image takes approximately three minutes. The X-ray dataset was created with 180 AP X-ray images of two different spine phantoms. It can be noticed that the number of images is not enough to train a deep-learning model, which is roughly estimated to require five thousand labeled samples to achieve acceptable results, but it requires millions of labeled samples to surpass human performance [140].

As it is highly time-demanding to increase the X-ray dataset size, another approach is taken for the CNN implementation. A set of 50.000 random images within the AP projection range are created using the DRR module and two DICOM volumes of the two available phantoms. The range of values differentiated by DoF is shown in Table 7-3. This created DRR dataset reduces the acquisition and labeling. The dataset is split into 60% for training, 20% for validating, and 20% for testing. Once the CNN is trained using the DRR training dataset, it is further fine-tuned using 144 of the X-ray images and tested using the remaining 36 X-ray images.

Table 7-3. Range of DRR parameters to create AP images

	Position			Orientation		
	X [mm]	Y [mm]	Z [mm]	X [degree]	Y [degree]	Z [degree]
Min value	-120	-60	-130	-15	165	-15
Max value	120	350	130	15	195	15

7.2.2. Development of the CNN Layout

The selected topology consists of three convolution layers followed by a max-polling layer per input. The feature maps of the three branches are combined by concatenation, such as the feature map size remains constant while the number of channels increases by three. In other words, the feature maps are stacked together. After the concatenation, three additional convolution and max-polling layers are applied. The last convolutional layer output is connected to the input of a FCNN. This FCNN consists of one input, three hidden, and one output layer. The CNN layout can be seen in Figure 7-5 and described in detail in Table 7-4. This network architecture is known as a multi-modal CNN due to the different nature of used images (modalities) for the inputs of the network. Nevertheless, the general CNN name is kept during this section to refer to this selected layout.

Table 7-4. Initial pose generator CNN topology

Layer	Type	(Out)-Size	# Channels	Kernel Size	Stride	Padding	Activation
Input x3	Image	512x512	1	-	-	-	-
2 x3	Convolution	512x512	8	5x5	1	2	ReLu
3 x3	Max Pooling	264x264	8	2x2	1	2	ReLu
4 x3	Convolution	264x264	16	5x5	1	2	ReLu
5 x3	Max Pooling	128x128	16	2x2	1	2	ReLu
6 x3	Convolution	128x128	24	5x5	1	2	ReLu
7 x3	Max Pooling	64x64	24	2x2	1	2	ReLu
8	Concatenation	64x64	72	-	-	-	-
9	Convolution	64x64	96	5x5	1	2	ReLu
10	Max Pooling	32x32	96	2x2	1	2	ReLu
11	Convolution	32x32	120	5x5	1	2	ReLu
12	Max Pooling	16x16	120	2x2	1	2	ReLu
13	Convolution	16x16	144	5x5	1	2	ReLu
14	Max Pooling	8x8	144	2x2	1	2	ReLu
15	Convolution	8x8	168	5x5	1	2	ReLu
16	FCNN	1024	-	-	-	-	ReLu
17	FCNN	512	-	-	-	-	ReLu
18	FCNN	128	-	-	-	-	ReLu
Output	FCNN	6	-	-	-	-	Linear

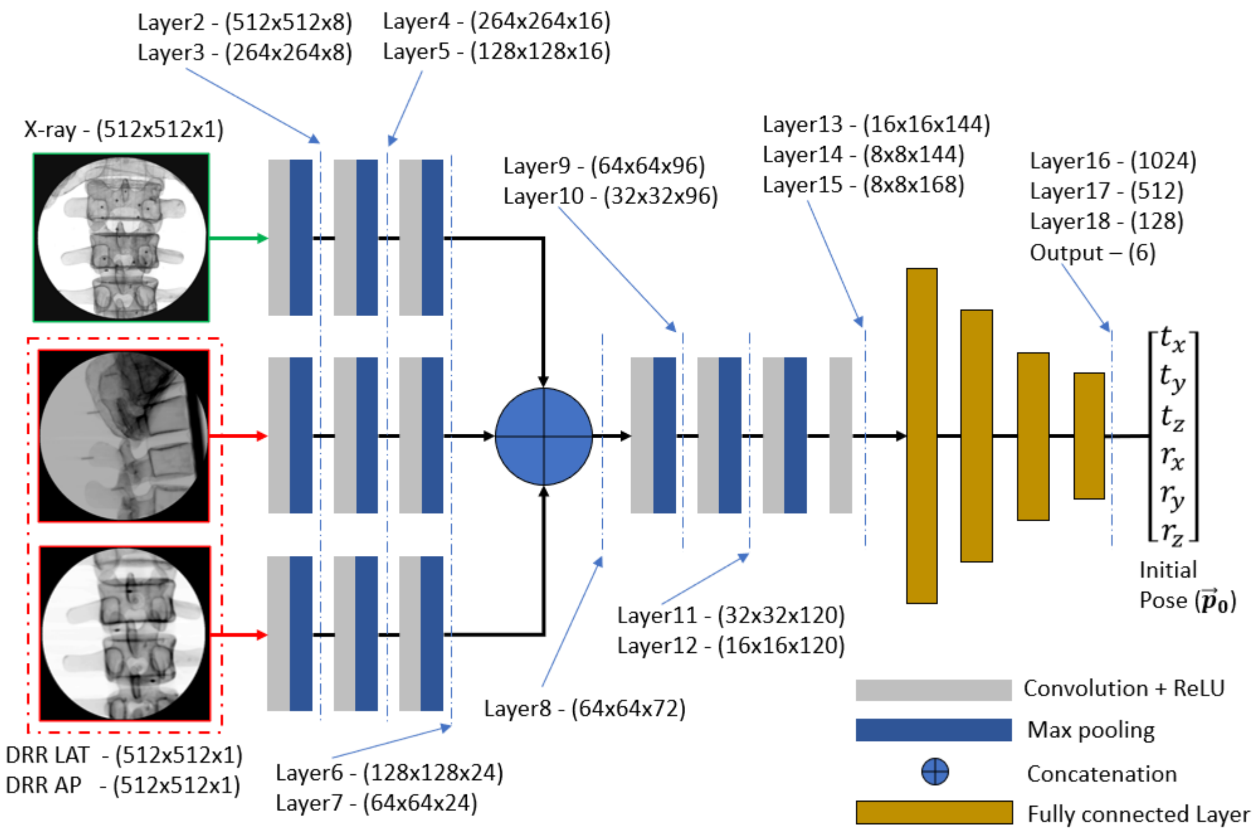

Figure 7-5. CNN Topology of the initial pose generator

The training is done using the DRR training dataset. The prediction accuracy and the cost function losses are calculated from the DRR training and the DRR validation dataset to control the training process. The training behavior is depicted in Figure 7-6. The loss function of both the training and validation stay very similar, indicating the model has a good fit. The training is stopped in epoch 200, where the validation and training loss stay at a plateau for more than 20 epochs.

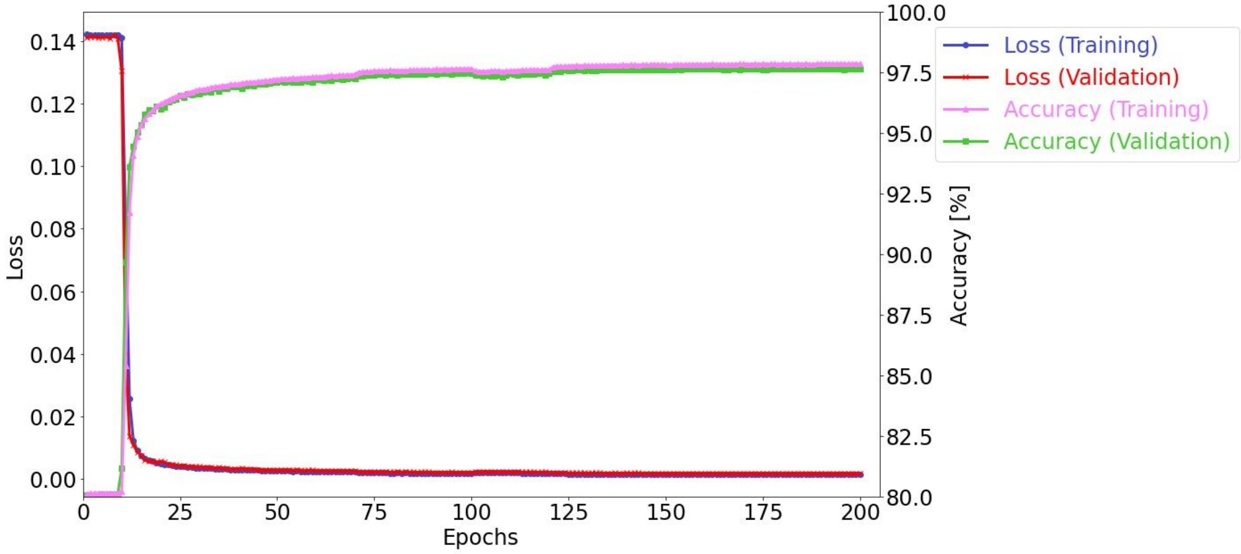

Figure 7-6. Training plot of the CNN approach for the initial pose generator

After training, the accuracy using the DRR testing dataset is calculated. The average accuracy is found to be 97.71%. Then the accuracy is once again computed but on the X-ray testing dataset. In the second test, the accuracy is found to be 89.53%. The detailed accuracy of the CNN on both datasets is presented in Table 7-5. The error of each DoF is calculated as the absolute error and normalized using the range of the elements (max value minus min value from Table 7-3). The accuracy and the error are expressed in equation *(138)*.

$$Error\_DoF_N = \frac{|actualDoF_N - predictedDoF_N|}{|maxValueDoF_N - minValueDoF_N|}$$
$$Accuracy\_DoF_N = \begin{cases} (1 - Error\_DoF_N) \cdot 100\% & if\ (1 - Error\_DoF_N) > 0 \\ 0 & Otherwise \end{cases} \tag{138}$$

The last column of Table 7-5 is the average prediction error of the six DoF.

Table 7-5. CNN accuracy on DRR and X-ray dataset

Dataset	Accuracy [%]						
	Position			Orientation			Average
	X	Y	Z	X	Y	Z	
DRR testing	98.22	97.93	97.82	97.27	97.60	97.42	97.71
X-ray testing	94.40	94.96	89.39	85.72	93.67	79.01	89.53

The CNN is fine-tuned using the X-ray training dataset after the training with the DRR dataset. There are different approaches to fine-tune a CNN regarding the number of layers to retrain [193]. Three different fine-tuning approaches regarding the retrained layers are tested: only the FCNN top layer (Figure 7-6 - layer16), the entire FCNN (Figure 7-6 – layers 16 to 18), and the entire CNN using the pre-trained weights.

7.2.3. Accuracy of the Initial Pose Generator Using CNN

For the fine-tuning procedure, the X-ray training dataset is used, and a small learning rate is set, i.e., $1e^{-5}$. The training plot, visible in Figure 7-7, depicts the accuracy and loss values of the training and validation X-ray datasets of the best-found approach, fine-tuning the entire pre-trained weights of the CNN. Similarly, as in the CNN training in section 7.1, an early stop mechanism is used. It shows that the best fit occurs around epoch 200 when the training and validation loss show similar values. After that, overfitting starts to occur since the loss on the training dataset keeps decreasing, and the validation loss remains nearly at the same level.

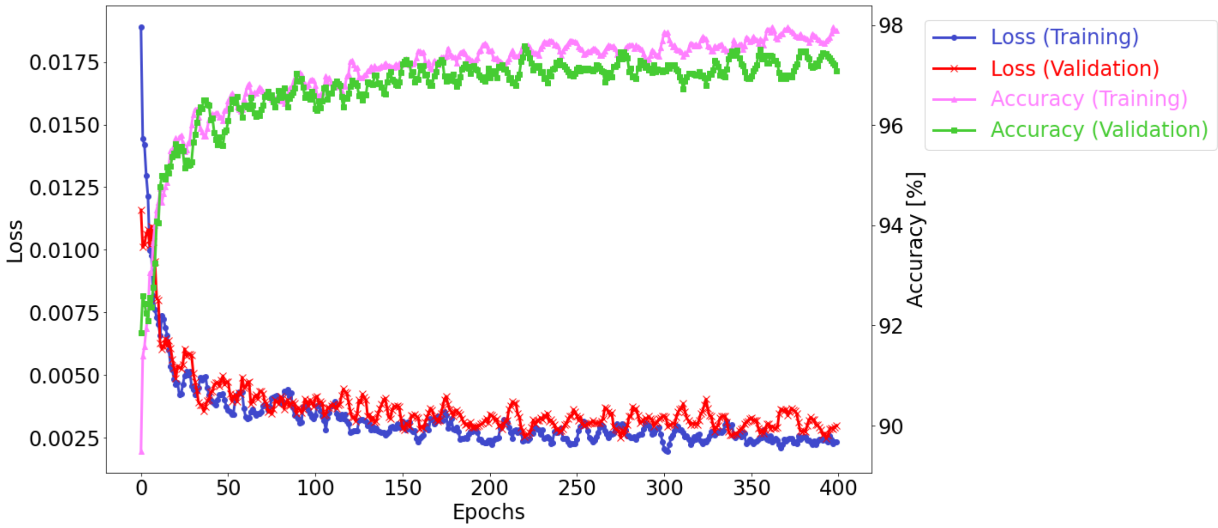

Figure 7-7. Fine-tuning training error of the CNN for the initial pose generator

The accuracies of the trained CNNs on the X-ray testing dataset using different fine-tuning methodologies are shown in Table 7-6. The individual pose components to be predicted, i.e., six DoF (three positions and rotations), can be seen in the first six columns. In the last column, the average of the six DoF is displayed.

Table 7-6. Accuracy results of the different fine-tuned CNNs

Fine-tuning	Accuracy [%]						
	Position			Orientation			Average
	X	Y	Z	X	Y	Z	
Upper layer FCNN	96.77	96.48	94.32	92.37	95.42	97.52	95.48
Entire FCNN	98.26	97.19	94.88	98.25	95.62	98.04	97.04
All CNN layers	97.21	97.57	96.29	97.64	95.94	97.95	97.09

The average accuracy of fine-tuning all CNN layers and fine-tuning only the FCNN are very similar, but the consistency of the accuracy along the six DoF is better when all CNN layers are fine-tuned. In Table 7-7, the accuracy of the fine-tuned CNN is given in absolute terms while inferring on the X-ray testing dataset. It can be noticed that the accuracies of the prediction when fine-tuning the weights on the entire CNN fall within the range, where the initial position produces reliable 2D/3D registration results, i.e., +/- 5 degrees and +/- 10mm in each axis [190]. Based on that behavior, the fine-tuning in all CNN layers is selected as the best approach.

Table 7-7. Accuracy of the initial pose generator discretized by parameter

Fine-tuning	Position accuracy			Orientation accuracy		
	X [mm]	Y [mm]	Z [mm]	X [degrees]	Y [degrees]	Z [degrees]
Upper layer FCNN	7.75	14.43	14.77	2.29	1.37	0.74
Entire FCNN	4.2	11.5	0.53	0.53	1.31	0.59
All CNN layers	6.70	9.96	9.64	0.71	1.22	0.62

It is worth noting that although this approach gives initial poses in an acceptable range for the 2D/3D registration process, its accuracy is not guaranteed in a real scenario as no human datasets are used to train the current CNN.

The results of this section show the feasibility of creating an automatic initial point generator in an intensity-based 2D/3D registration. More DRR images from CT-scans and X-rays coming from different patients could obtain more accurate results and guarantee the generalization of the solution.

7.2.4. Structure of the 2D/3D Registration Procedure Using the Initial Pose Generator

In section 4.7.4, the 2D/3D registration procedure is introduced and summarized in the block diagram depicted in Figure 4-27. The developed initial pose generator is integrated into the registration procedure. The 2D/3D registration requires only the user to load the X-ray images and the DICOM volume; after that, the registration process initiates by running the CNN to infer the initial pose. Based on a i7-8700K CPU at 3.7Ghz and a Nvidia GTX 1080 GPU, the runtime for loading the CNN in the memory is about four seconds, while inferencing is executed in less than one second. The initial pose generator requires five seconds in total. Once the prediction is acquired, the 2D/3D registration process starts, giving the registration pose. The updated block diagram of the 2D/3D registration can be seen in Figure 7-8.

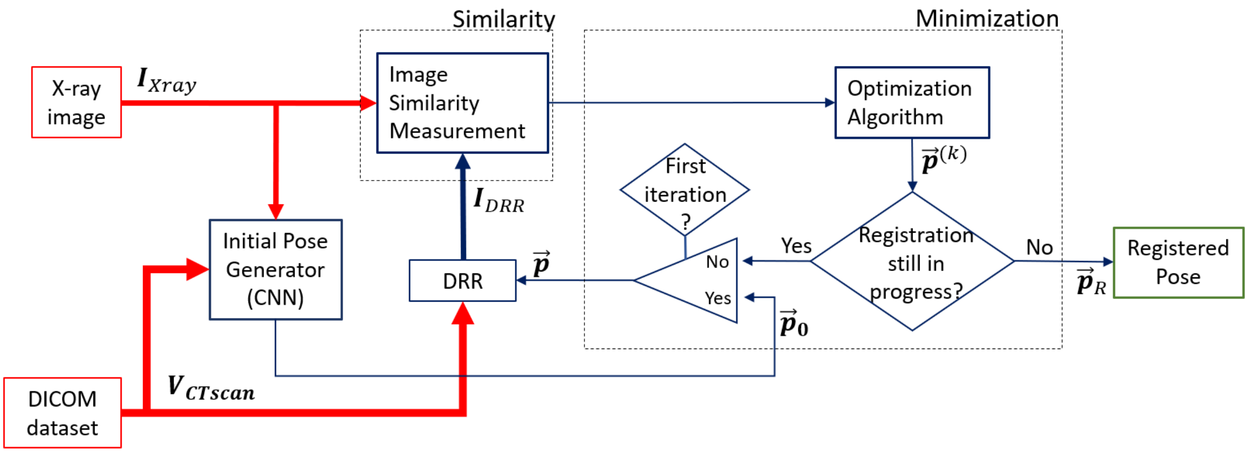

Figure 7-8. Updated block diagram of the 2D/3D registration with initial pose generator

The incorporation of the CNN for predicting the initial pose for the 2D/3D registration constitutes the last scaffold to achieve a fully automatic local 2D/3D registration. This achievement fulfills the automation objective, proposed as one of the primary outcomes of this work.

7.2.5. Discussion

A deep-learning approach for the initial pose generator is conceived using three images as inputs and six outputs: The AP X-ray image to be registered and two DRR images, AP and LAT projection of the DICOM volume used for the registration. The outputs are the three positions and three rotations that composed the ${}^{DRR}\boldsymbol{T}_{DICOM}$ pose of the DRR module in vector form, $\vec{\boldsymbol{p}} \in \mathbb{R}^6$, used in the 2D/3D registration for the initial pose.

The training is carried out in two stages. The CNN is initially trained using a large dataset based on random AP DRR images. In the second stage, the CNN is fine-tuned using a small dataset of X-ray images taken with a C-arm. This approach is taken due to the inviability of taken a large dataset of actual X-ray images. The results of the implemented CNN show that the achieved predictions fit the conditions of an acceptable initial pose for the 2D/3D registration, i.e., +/- 5 degrees and +/- 10mm in each axis [190] within an inference time of about five seconds.

With the obtained results, the implemented CNN can be considered to be used as the initial pose generator for the 2D/3D registration, but as no X-ray and CT-scan images were coming from patients, further evaluation is still required.

Additionally, the 2D/3D registration block diagram is updated, as Figure 7-4 shows. In the new structure, the user interaction is reduced to input the X-ray images and the DICOM volume. After the inputs are set, the 2D/3D registration runs autonomously until the registration pose is obtained. This updated process constitutes an automatic 2D/3D registration, which is one of the main objectives to be achieved in this work.

8. Conclusions and Final Discussion

8.1. Conclusions and Discussion

The main task of this thesis is the implementation of a fully automatic intensity-based 2D/3D registration using 3D pre-operative CT-scans modalities (in DICOM format) and 2D intra-operative X-ray modalities. In practice, it means to find a transformation matrix, ${}^{ARB}\boldsymbol{T}_{DICOM}$, that converts a planned pose, usually the pose of an implant, in the frame of the pre-operative data, {DICOM}, into a reference frame fixed to the patient {ARB}. The implant pose is written as a transformation matrix, ${}^{DICOM}\boldsymbol{T}_{IMPLANT}$ the planned pose in the pre-operative modality, and ${}^{ARB}\boldsymbol{T}_{IMPLANT}$ the planned pose in the patient frame. The surgical robot end-defector {TRB} and {ARB} are tracked by the navigation system, allowing the transformation between the implant pose and the surgical robot system.

An intermediate 2D/3D registration result is the pose in the digitally reconstructed radiograph (DRR) module with respect to the DICOM volume, ${}^{DRR}\boldsymbol{T}_{DICOM}$. A set of mathematical transformations is determined, moving ${}^{DRR}\boldsymbol{T}_{DICOM}$ into the useful ${}^{ARB}\boldsymbol{T}_{DICOM}$, which is the bridge connecting the pre-operative planned data into the operating scenario. The planned data is transformed to {ARB}, a frame known for the navigated robot, where it can reach a target. Then ${}^{ARB}\boldsymbol{T}_{DICOM}$ closes the loop of making available pre-operative planned data to the surgical robot.

The C-arm is mathematically defined with the pinhole camera model, which describes the C-arm with two matrices: the intrinsic and extrinsic parameter matrices. The characterization is done using the direct linear transformation algorithm [31] and a parametrization device that contains twelve steel beads. The characterization determines important C-arm features like the source-detector distance and the transformation from the source to the characterization device. A well-calibrated device containing seven landmarks is used to test the accuracy of the C-arm parametrization. The expected and measured landmarks show an average error of 2.5 pixels based on the ten X-ray images. The used C-arm, a Ziehm Vario 3D, has an image pixel spacing of 0.365mm/px. The average error is further expressed as 0.9125mm, indicating an accurate C-arm parametrization.

An undistortion process for X-ray images coming from C-arms with image intensifier technology is implemented. An undistortion plate with 60 fiducials is manufactured and installed on the C-arm detector. Using a mapping function created by a fifth order bi-polynomial regression, the X-ray images can be freed of distortion by an image warping process [159]. A further approach is introduced, removing the fiducials from the X-ray image using an inpainting algorithm [163].

A fast DRR module, based on the Siddon-Jacobs algorithm [104], is implemented and later optimized using parallel computing. The parallel implementation uses the Nvidia CUDA libraries, taking advantage of the hundreds of kernels available in the computer graphics card. The average DRR rendering time is 10ms in comparison with an average rendering time of 2.5 seconds using the conventional implementation. The test is performed creating 568x568 pixels images using a DICOM of 512x512x442 voxels.

The implemented 2D/3D intensity-based registration uses an iterative process to find the transformation between a 3D volume and a 2D image process, ${}^{DRR}\boldsymbol{T}_{DICOM}$. A CT-scan is projected as a 2D image with the DRR module. The DRR image is compared with an X-ray image using a merit function, image similarity measurement. An optimizer is used to find the CT-scan pose that matches the projection of the 3D image with the 2D image. This iterative process keeps creating DRR images on new poses suggested by the optimization algorithm. The loop *DRR - merit function - optimization* breaks when the merit function reaches its minimum value, meaning the registration is over.

An evaluation of image similarity measurements is carried out. The accuracy showed by mutual information MI is the highest among the other tested image similarity measurements, determining that (MI) is the most suitable measure for the implemented registration process. Likewise, an evaluation of different optimization algorithms is performed, finding that best neighbors (BN) leads the registration process to the smallest registration errors. A total of 100 registrations and 200 planned screws are analyzed to find the implant accuracy. The combination of MI, BN, and the inclusion of a manually selected region of interest (ROI) in a single lumbar vertebra produces a 2D/3D registration with an implant accuracy of 1.49 ± 0.48mm within a confidence interval of 68.27%. This accuracy value is found using an approach based solely on the navigation system, i.e., the surgical robot system is not involved. An advantage of limiting the computation of the accuracy using only the navigation system is to avoid the risk that the robot can introduce some errors.

It is required to give an initial pose to the registration procedure previously to its execution. There is a direct correlation between the accuracy of the registration result and the closeness of the initial pose to the actual true registration pose, ±5 degrees and ±10mm in each axis at most [190]. As the initial pose consists of six DoF, it is no intuitive for a user to guess and insert this value manually. A graphical approach is developed, showing the overlapping of the X-ray image and a DRR image created from the same rendered view of the DICOM volume.

The initial pose selection is further improved, such as the pose value to the registration process is not selected manually but inferred from a CNN. This initial pose generator reduces the user intervention with the software during the surgical intervention and guarantees the initial pose closeness with the actual registration pose.

An optimization in the fiducial detection for the undistortion of X-ray images from conventional C-arms is developed using a deep-learning approach. A fully automatic undistortion procedure is introduced, having a CNN as the fiducial detection unit. Each fiducial center is predicted with a maximum absolute error of 0.42 pixels.

8.2. Future Work

The proposed objectives for this thesis are achieved, but there is still room for improvement and development of additional functionalities. The following paragraphs describe future works that can be carried out.

It is noticed on the X-ray images used for the registration process that the structure of the C-arm parametrization device leaves footprints, affecting the registration accuracy. Consequently, running an inpainting process to remove the structure could be considered, but a better approach is remanufacturing the parametrization device using a more translucid X-ray material instead of the 3D-Printing material.

One of the approaches that makes the 2D/3D registration accuracy better is the ROI, the manual selection and definition of the X-ray image boundaries. An automatic vertebra segmentation for X-ray images can be developed to define several ROIs without user intervention.

Likewise, implementing a DICOM vertebra segmentation process could automate the creation of several sub-volumes. Each sub-volume could be matched and registered with the different ROIs in the X-ray images. Combining the vertebra segmentation in X-ray images and DICOM volumes would lead to developing a fully automated quasi-global 2D/3D registration.

The current initial pose generator is trained using two different phantom models. Although the results are promising, this initial pose generator cannot be considered yet to be used in a real scenario since its behavior in actual patients has not been tested. Further training using more DICOM volumes is necessary to create a robust generator and guarantee the CNN generalization.

Appendix A. Derivation of Formulas, Procedures, and Functions

A.1 Transformation between Rotation Matrices and Quaternions

Let $\mathbf{q} \epsilon \mathbb{H}$ be a quaternion defined as $\mathbf{q} = (s, v_x, v_y, v_z)$, and $\boldsymbol{R} \in \mathbb{R}^{3\times3}$ be a rotation matrix,

where $\boldsymbol{R} = \begin{bmatrix} r_{11} & r_{12} & r_{13} \\ r_{21} & r_{22} & r_{23} \\ r_{31} & r_{32} & r_{33} \end{bmatrix}$

The elements of the rotation matrix with respect to a normalized quaternion, i.e., r_{11} to r_{33} in terms of s, v_x, v_y, v_z are expressed in *(139)*.

$$\begin{aligned} r_{11} &= s^2 + {v_x}^2 - {v_y}^2 - {v_z}^2 \\ r_{12} &= 2 \cdot \left((v_x \cdot v_y) - (s \cdot v_z) \right) \\ r_{13} &= 2 \cdot \left((v_x \cdot v_z) + (s \cdot v_y) \right) \\ r_{21} &= 2 \cdot \left((v_x \cdot v_y) + (s \cdot v_z) \right) \\ r_{22} &= s^2 - {v_x}^2 + {v_y}^2 - {v_z}^2 \\ r_{23} &= 2 \cdot \left((v_y \cdot v_z) - (s \cdot v_x) \right) \\ r_{31} &= 2 \cdot \left((v_x \cdot v_z) - (s \cdot v_y) \right) \\ r_{32} &= 2 \cdot \left((v_y \cdot v_z) + (s \cdot v_x) \right) \\ r_{33} &= s^2 - {v_x}^2 - {v_y}^2 + {v_z}^2 \end{aligned} \tag{139}$$

The elements of the rotation matrix with respect to the quaternions, i.e., r_{11} to r_{33} in terms of s, v_x, v_y, v_z are expressed in *(140)*.

$$\begin{aligned} s &= \frac{1}{2}\sqrt{r_{11} + r_{22} + r_{33} + 1} \\ v_x &= sign(r_{32} - r_{23}) \cdot \frac{1}{2}\sqrt{r_{11} - r_{22} - r_{33} + 1} \\ v_y &= sign(r_{13} - r_{31}) \cdot \frac{1}{2}\sqrt{-r_{11} + r_{22} - r_{33} + 1} \\ v_z &= sign(r_{21} - r_{12}) \cdot \frac{1}{2}\sqrt{-r_{11} - r_{22} + r_{33} + 1} \end{aligned} \tag{140}$$

The demonstration of *(139)* and *(140)* is out of the scope of this work, but it can be found in [194].

A.2 Eigenvalues and Eigenvectors

Let $\mathbf{A}$ be an $n \times n$ matrix of real components, λ be a scalar in $\mathbb{C}$, and $\vec{\boldsymbol{v}}$ a vector in $\mathbb{C}^n$ such that

$$\boldsymbol{A} \cdot \vec{\boldsymbol{v}} = \lambda \cdot \vec{\boldsymbol{v}} \tag{141}$$

The vector $\vec{\boldsymbol{v}} \neq \vec{\mathbf{0}}$ is called an eigenvector of $\boldsymbol{A}$ corresponding to the scalar λ, also known as an eigenvalue. *(141)* can be rewritten as *(142)* following the consideration that there is a $\vec{\boldsymbol{v}} \neq \vec{\mathbf{0}}$.

$$(\boldsymbol{A} - \lambda \cdot \boldsymbol{I}) \cdot \vec{\boldsymbol{v}} = \vec{\mathbf{0}} \tag{142}$$

The non-trivial solution in (*142*), i.e., $\vec{\boldsymbol{v}} \neq \vec{\mathbf{0}}$, requires that $(\boldsymbol{A} - \lambda \cdot \boldsymbol{I})$ is non-invertible. It means that

$$p(\lambda) = det(\boldsymbol{A} - \lambda \cdot \boldsymbol{I}) = 0 \tag{143}$$

$p(\lambda)$ is known as the characteristic polynomial of $\boldsymbol{A}$ and has n roots. The roots of $p(\lambda)$ make the set of eigenvalues $\{\lambda_1, \dots, \lambda_n\}$ of the matrix $\boldsymbol{A}$.

The eigenvector $\vec{\boldsymbol{v}}_{\boldsymbol{i}}$ is found solving the resulting homogenous system from *(142)* using the corresponding eigenvalue λ_i. In the end the set of eigenvectors of the matrix $\boldsymbol{A}$ is formed by the n linearly independent vectors, $\{\vec{\boldsymbol{v}}_1, \dots, \vec{\boldsymbol{v}}_n\}$ [195].

A.3 Singular Value Decomposition (SVD)

SVD is a useful matrix decomposition, which is commonly applied in the solution of over-determined system of equations. Let $\mathbf{A}$ be an $m \times n$ matrix with $m \geq n$. The SVD of $\boldsymbol{A}$ can be expressed as:

$$\boldsymbol{A} = \boldsymbol{U}_h \cdot \boldsymbol{\Sigma} \cdot \boldsymbol{V}_b^T \tag{144}$$

where $\boldsymbol{U}_h \in \mathbb{R}^{m \times m}$ and $\boldsymbol{V}_b^T \in \mathbb{R}^{n \times n}$ are orthonormal matrices, and $\boldsymbol{\Sigma} \in \mathbb{R}^{m \times n}$ is a non-negative diagonal matrix. The demonstration of equation *(144)* can be seen in [196].

The diagonal values in $\mathbf{\Sigma}$ are the singular values of $\boldsymbol{A}$ arranged in descendent order. The singular values of $\boldsymbol{A}$ must not be confused with the eigenvalues of $\boldsymbol{A}$. The singular values of $\boldsymbol{A}$ are the square roots of eigenvalues from or $\boldsymbol{A}^{\mathrm{T}} \cdot \boldsymbol{A}$. Let $\vec{\boldsymbol{\sigma}} = [\sigma_1, \dots, \sigma_n]^T$ be the vector composed by the eigenvalues of $\boldsymbol{A}^T \cdot \boldsymbol{A}$, such as $\sigma_1 \geq \sigma_2 \geq \cdots \geq \sigma_{\mathrm{n}} \geq 0$. $\mathbf{\Sigma}$ can be written as:

$$\boldsymbol{\Sigma} = \begin{bmatrix} \sigma_1 & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & \sigma_{\boldsymbol{n}} \\ & \vdots & \\ & 0 & \end{bmatrix} \tag{145}$$

The columns of $\boldsymbol{U}_h$ are composed by the eigenvector of the matrix $\boldsymbol{A} \cdot \boldsymbol{A}^{\mathrm{T}}$, and the eigenvectors of $\boldsymbol{A}^{\mathrm{T}} \cdot \boldsymbol{A}$ make the columns of $\boldsymbol{V}_b$. let $\mathbf{H} = \{\vec{\boldsymbol{h}}_1, \dots, \vec{\boldsymbol{h}}_{\mathrm{m}} \mid \vec{\boldsymbol{h}}_i \in \mathbb{R}^m \}$ be the set of eigenvectors of $\boldsymbol{A} \cdot \boldsymbol{A}^{\mathrm{T}}$ and $\mathbf{B} = \{\boldsymbol{b}_1, \dots, \boldsymbol{b}_{\mathrm{n}} \mid \boldsymbol{b}_j \in \mathbb{R}^{\boldsymbol{n}}\}$ be the set of eigenvectors of $\boldsymbol{A}^{\mathrm{T}} \cdot \boldsymbol{A}$, the matrices $\mathbf{U}_{\mathrm{h}}$ and $\boldsymbol{V}_b$ are built as follows:

$$\boldsymbol{U}_h = [\boldsymbol{h}_1 \quad \cdots \quad \boldsymbol{h}_m] \tag{146}$$

$$\boldsymbol{V}_b = [\boldsymbol{b}_1 \quad \cdots \quad \boldsymbol{b}_n] \tag{147}$$

The calculations of the eigenvalues and eigenvectors of a matrix can be seen in appendix A.2.

A.4 RQ Simplification

Assuming orthogonal forward rotation matrices in space $\mathbb{R}^3$. It is possible to define one matrix per axis.

$$\boldsymbol{Rot_x}(\theta) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos(\theta) & -\sin(\theta) \\ 0 & \sin(\theta) & \cos(\theta) \end{bmatrix}$$

$$\boldsymbol{Rot_y}(\beta) = \begin{bmatrix} \cos(\beta) & 0 & \sin(\beta) \\ 0 & 1 & 0 \\ -\sin(\beta) & 0 & \cos(\beta) \end{bmatrix} \tag{148}$$

$$\boldsymbol{Rot_z}(\gamma) = \begin{bmatrix} \cos(\gamma) & -\sin(\gamma) & 0 \\ \sin(\gamma) & \cos(\gamma) & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

Before starting, we can define the matrix $\boldsymbol{A}$ as:

$$\boldsymbol{A} = \begin{bmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{bmatrix} \tag{149}$$

Matrix $\boldsymbol{A}$ will be decomposed, such as $\boldsymbol{A} = \boldsymbol{R} \cdot \boldsymbol{Q}$, where $\boldsymbol{R}$ is a right-triangular matrix, and $\boldsymbol{Q}$ is an orthogonal matrix. The matrix $\boldsymbol{Q}$ will be then a product $\boldsymbol{Rot_x(\theta)} \cdot \boldsymbol{Rot_y(\beta)} \cdot \boldsymbol{Rot_z(\gamma)}$. Then the matrix $\mathbf{A}$ can be calculated as:

$$\boldsymbol{A} = \boldsymbol{R} \cdot \boldsymbol{Rot_z}(\gamma)^T \cdot \boldsymbol{Rot_y}(\beta)^T \cdot \boldsymbol{Rot_x}(\theta)^T \tag{150}$$

Each of these rotations will make zero the lower half of the matrix to build the right-triangular matrix

$$\boldsymbol{A} \cdot \boldsymbol{Rot_x}(\theta) \cdot \boldsymbol{Rot_y}(\beta) \cdot \boldsymbol{Rot_z}(\gamma) = \boldsymbol{R} \tag{151}$$

Then, the rotation with x is:

$$\boldsymbol{A} \cdot \boldsymbol{Rot_x}(\theta) = \begin{bmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{bmatrix} \cdot \begin{bmatrix} 1 & 0 & 0 \\ 0 & cos(\theta) & -sin(\theta) \\ 0 & sin(\theta) & cos(\theta) \end{bmatrix} = \boldsymbol{B} \tag{152}$$

where

$$\boldsymbol{B} = \begin{bmatrix} a_{11} & b_{12} & b_{13} \\ a_{21} & b_{22} & b_{23} \\ a_{31} & b_{32} & b_{33} \end{bmatrix} \tag{153}$$

It can be seen that the entire column 1 of $\boldsymbol{B}$ remains unchanged, but the element b_{32} will be:

$$b_{32} = a_{32}cos(\theta) + a_{33}sin(\theta) \tag{154}$$

To make the matrix $\boldsymbol{R}$ a right triangular, b_{32} must be zero:

$$a_{32}cos(\theta) + a_{33}sin(\theta) = 0 \tag{155}$$

Solving this equation can be found that:

$$\theta_1 = \tan^{-1}(-\frac{a_{32}}{a_{31}})$$
$$\theta_2 = tan^{-1}(-\frac{a_{32}}{a_{31}}) + \pi \tag{156}$$

This duality can be resolved by making the diagonal of the right-triangular matrix positive. Following a similar procedure, the rotation in y is applied

$$\boldsymbol{A} \cdot \boldsymbol{Rot_x}(\theta) \cdot \boldsymbol{Rot_y}(\beta) = \boldsymbol{B} \cdot \boldsymbol{Rot_y}(\beta) = \begin{bmatrix} a_{11} & b_{12} & b_{13} \\ a_{21} & b_{22} & b_{23} \\ a_{31} & 0 & b_{33} \end{bmatrix} \cdot \begin{bmatrix} cos(\beta) & 0 & sin(\beta) \\ 0 & 1 & 0 \\ -sin(\beta) & 0 & cos(\beta) \end{bmatrix} = \boldsymbol{C} \tag{157}$$

Where

$$\boldsymbol{C} = \begin{bmatrix} c_{11} & b_{12} & c_{13} \\ c_{21} & b_{22} & c_{23} \\ c_{31} & 0 & c_{33} \end{bmatrix} \tag{158}$$

Following a similar procedure, the element c_{31} will be made zero, with the selection of the right β, which follows the next relations:

$$\beta_1 = \tan^{-1}\left(\frac{b_{31}}{b_{33}}\right)$$
$$\beta_2 = tan^{-1}\left(\frac{b_{31}}{b_{33}}\right) + \pi \tag{159}$$

Again, the selection of the angle is given by having a positive diagonal. Finally, the rotation in z is applied:

$$\boldsymbol{A} \cdot \boldsymbol{Rot_x}(\theta) \cdot \boldsymbol{Rot_y}(\beta) \cdot \boldsymbol{Rot_z}(\gamma) = \boldsymbol{B} \cdot \boldsymbol{Rot_y}(\beta) \cdot \boldsymbol{Rot_z}(\gamma) = \boldsymbol{C} \cdot \boldsymbol{Rot_z}(\gamma) = \boldsymbol{R} \tag{160}$$

$$\boldsymbol{R} = \begin{bmatrix} c_{11} & b_{12} & c_{13} \\ c_{21} & b_{22} & c_{23} \\ 0 & 0 & c_{33} \end{bmatrix} \cdot \begin{bmatrix} cos(\gamma) & -sin(\gamma) & 0 \\ sin(\gamma) & cos(\gamma) & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{161}$$

Where

$$\boldsymbol{R} = \begin{bmatrix} r_{11} & r_{12} & c_{13} \\ r_{21} & r_{22} & c_{23} \\ 0 & 0 & c_{33} \end{bmatrix} \tag{162}$$

With a similar procedure, the element r_{21} will be made zero, with the selection of the right γ, which follows the next relations:

$$\gamma_1 = \tan^{-1}(-\frac{c_{21}}{c_{22}}) \\ \gamma_2 = tan^{-1}(-\frac{c_{21}}{c_{22}}) + \pi \tag{163}$$

After this final multiplication

$$\boldsymbol{R} = \begin{bmatrix} r_{11} & r_{12} & r_{13} \\ 0 & r_{22} & r_{23} \\ 0 & 0 & r_{33} \end{bmatrix} \tag{164}$$

Which is a right-triangular matrix.

The $\mathbf{Q}$ matrix can be found as a product of orthogonal rotation matrices [31]:

$$\boldsymbol{Q} = \boldsymbol{Rot_z}(\gamma)^{\boldsymbol{T}} \cdot \boldsymbol{Rot_y}(\beta)^{\boldsymbol{T}} \cdot \boldsymbol{Rot_x}(\theta)^{\boldsymbol{T}} \tag{165}$$

$$\boldsymbol{Q} = \begin{bmatrix} cos(\gamma) & sin(\gamma) & 0 \\ -sin(\gamma) & cos(\gamma) & 0 \\ 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} cos(\beta) & 0 & -sin(\beta) \\ 0 & 1 & 0 \\ sin(\beta) & 0 & cos(\beta) \end{bmatrix} \cdot \begin{bmatrix} 1 & 0 & 0 \\ 0 & cos(\theta) & sin(\theta) \\ 0 & -sin(\theta) & cos(\theta) \end{bmatrix} \tag{166}$$

A.5 Least Squares Solution

Let the system of equation $\boldsymbol{A} \cdot \vec{\boldsymbol{x}} = \vec{\boldsymbol{b}}$ be formed by m equations and n unknowns, i.e., $\boldsymbol{A} \in \mathbb{R}^{m \times n}$, $\vec{\boldsymbol{x}} \in \mathbb{R}^n$, and $\vec{\boldsymbol{b}} \in \mathbb{R}^m$. Let be assumed that $m > n$, meaning the system does not have an exact solution. If the columns of $\boldsymbol{A}$ are linearly independent, i.e., rank of $\boldsymbol{A}$ is n, an approximation for $\vec{\boldsymbol{x}}$ consists of determining the least square solution, such as the error expressed by $\|\boldsymbol{A} \cdot \vec{\boldsymbol{x}} - \vec{\boldsymbol{b}}\|$ is minimized.

The error can be expressed using SVD, see appendix A.3, as $\|\boldsymbol{A} \cdot \vec{\boldsymbol{x}} - \vec{\boldsymbol{b}}\| = \|\boldsymbol{U}_h \cdot \boldsymbol{\Sigma} \cdot \boldsymbol{V}_b^T \cdot \vec{\boldsymbol{x}} - \vec{\boldsymbol{b}}\|$. Since the norm is preserved in orthogonal transforms, then $\|\boldsymbol{U}_h \cdot \boldsymbol{\Sigma} \cdot \boldsymbol{V}_b^T \cdot \vec{\boldsymbol{x}} - \vec{\boldsymbol{b}}\| = \|\boldsymbol{\Sigma} \cdot \boldsymbol{V}_b^T \cdot \vec{\boldsymbol{x}} - \boldsymbol{U}_h^T \cdot \vec{\boldsymbol{b}}\|$. Now let the auxiliary vectors $\vec{\boldsymbol{y}}$ and $\vec{\boldsymbol{b}}'$ be defined as $\vec{\boldsymbol{y}} = \boldsymbol{V}_b^T \cdot \vec{\boldsymbol{x}}$ and $\vec{\boldsymbol{b}}' = \boldsymbol{U}_h^T \cdot \vec{\boldsymbol{b}}$, then the minimization problem is expressed in *(167)* as $\|\boldsymbol{\Sigma} \cdot \vec{\boldsymbol{y}} - \vec{\boldsymbol{b}}'\|$.

$$\begin{bmatrix} \sigma_1 & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & \sigma_n \\ & \vdots & \\ & 0 & \end{bmatrix} \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_n \end{bmatrix} = \begin{bmatrix} b_1' \\ \vdots \\ b_n' \\ \vdots \\ b_m' \end{bmatrix} \tag{167}$$

The closest that $\boldsymbol{\Sigma} \cdot \vec{\boldsymbol{y}}$ approaches $\vec{\boldsymbol{b}}'$ is described by the vector in equation *(168)* [31]:

$$\vec{\boldsymbol{b}}' = [b_1' \quad \cdots \quad b_n' \quad 0 \quad \cdots \quad 0]^T \tag{168}$$

From *(167)* and *(168)*, it can be found that every component of $\vec{\boldsymbol{y}}$ can be set as $y_i = {b'_i}/{\sigma_i}$ for $i \in \{1, \dots, n\}$. Once $\vec{\boldsymbol{y}}$ is calculated, the least square solution $\vec{\boldsymbol{x}}$ can be retrieved as follows:

$$\vec{\boldsymbol{x}} = \boldsymbol{V}_b \cdot \vec{\boldsymbol{y}} \tag{169}$$

A.6 Least Squares Solution of Homogeneous System of Linear Equations

Let the system of equation $\boldsymbol{A} \cdot \vec{\boldsymbol{x}} = \vec{\boldsymbol{0}}$ be formed by m equation and n unknowns, meaning the system is overdetermined. The trivial solution $\vec{\boldsymbol{x}} = \vec{\boldsymbol{0}}$ is discarded; in other words, a non-zero solution is desired, which is commonly non-unique. Due to the equality to zero of the system of equations, if $\vec{\boldsymbol{x}}$ is a solution of the system, it would be $\vec{\boldsymbol{x}}$ multiplied by a constant k. The minimization problem can be then restricted in the following way:

Find the $\vec{\boldsymbol{x}}$ that minimizes $\|\boldsymbol{A} \cdot \vec{\boldsymbol{x}}\|$ with $\|\vec{\boldsymbol{x}}\| = 1$. Using SVD, as explain in appendix A.3, the minimization problem is transformed to minimize $\|\boldsymbol{U}_h \cdot \boldsymbol{\Sigma} \cdot \boldsymbol{V}_a^T \cdot \vec{\boldsymbol{x}}\|$. The demonstration of the result can be seen in [31]. The minimization solution to the homogenous system of linear equation, $\vec{\boldsymbol{x}}$, is given by the last column of $\boldsymbol{V}_a$, which is the eigenvector of $\boldsymbol{A}^{\mathrm{T}} \cdot \boldsymbol{A}$ corresponding to the smallest eigenvalue.

A.7 R-squared and Standard Error of the Linear Regression

The standard error of the regression (S) and R-squared (R^2) are two important goodness of fit measurement to evaluate the fit of a (linear) regression. The standard error of the regression gives the average error (distance) between the actual values and the estimated regression function. The units of the standard error remain the same as the actual values. R-squared gives a fitting value from 0 to 1 (0-100%) based on the actual values scattering around the estimated regression function. A High R-squared value, e.g., 0.99, indicates a small difference between the actual values and the estimated regression function; in other words, a good fit.

To give the equations of the standard error of the regression and the R-squared, the following definitions are required. Let N be the number of available samples in the set $\mathbf{X}$ and $\mathbf{Y}$, such as $\mathbf{X} = \{\mathrm{x}_1, \mathrm{x}_2, \dots, \mathrm{x}_N\}$ and $\mathbf{Y} = \{\mathrm{y}_1, \mathrm{y}_2, \dots, \mathrm{y}_N\}$. Every element of the sets $\mathbf{X}$ and $\mathbf{Y}$ belong to the set of real numbers, i.e., $\mathrm{x}_i, \mathrm{y}_i \in \mathbb{R}$. Let $f(x)$ be the linear regression function calculated from the sets $\mathbf{X}$ and $\mathbf{Y}$, such as $f: \mathbf{X} \rightarrow \mathbf{Y}$. The standard error can be calculated using the equation *(170)* , and R-squared with equation *(171)*. The demonstrations of these equations are out of this work scope, but they can be found in [198].

$$S = \sqrt{\frac{1}{N-1} \cdot \sum_{i=1}^{N} \left(y_i - f(y_i)\right)^2} \tag{170}$$

$$R^2 = 1 - \frac{\sum_{i=1}^{N} \left(y_i - f(y_i)\right)^2}{\sum_{i=1}^{N} (y_i - \bar{Y})^2} \tag{171}$$

where $\overline{\mathrm{Y}}$ is the average of the set $\mathbf{Y}$.

A.8 Stereo Lithography (STL) to DICOM format conversion

A 2D or 3D modality in DICOM format follows the standard output imaging format of a medical device, e.g., C-arm and CT-scan. For research purposes, in this case 2D/3D registration, is useful to design devices using computer-aided design (CAD) software and then accurately produce them using computer numerical control (CNC) machines or 3D-printers. To incorporate such devices in a medical framework, it is necessary to capture the 3D volume of the produced element using a CT-scan or an MRI. The accessibility to such medical modalities is restricted, but having the CAD model in standard stereolithography (STL) format gives enough information to convert it into a DICOM format.
To start, the basics about the DICOM structure are clarified. Later the basics of the STL structure are explained. Finally, the conversion from the STL structure to the DICOM structure is elaborated.

DICOM Format Structure

In sections 4.6.1 and 4.7.1, it is explained that the DICOM contains the information in a 3D array using image slices forming X-Y planes and stacking in the Z direction. The physical distance between pixels in the X-Y image is given by the DICOM tag pixel spacing (mm/px). The DICOM tag "spacing between slices" (mm/frame) gives the distance between planes in the Z-direction. The lower the pixels spacing and the spacing between slices, the more quality the DICOM volume has, e.g., 0.2 mm/px (mm/frame). Having the volume dimension in pixels units, it is possible to find its physical dimension. For conversion purposes, the quality of the DICOM and the final dimensions are user-inputs. The physical dimensions are constrained as they come from the STL file. The previous ideas are illustrated in Figure A-8-1.

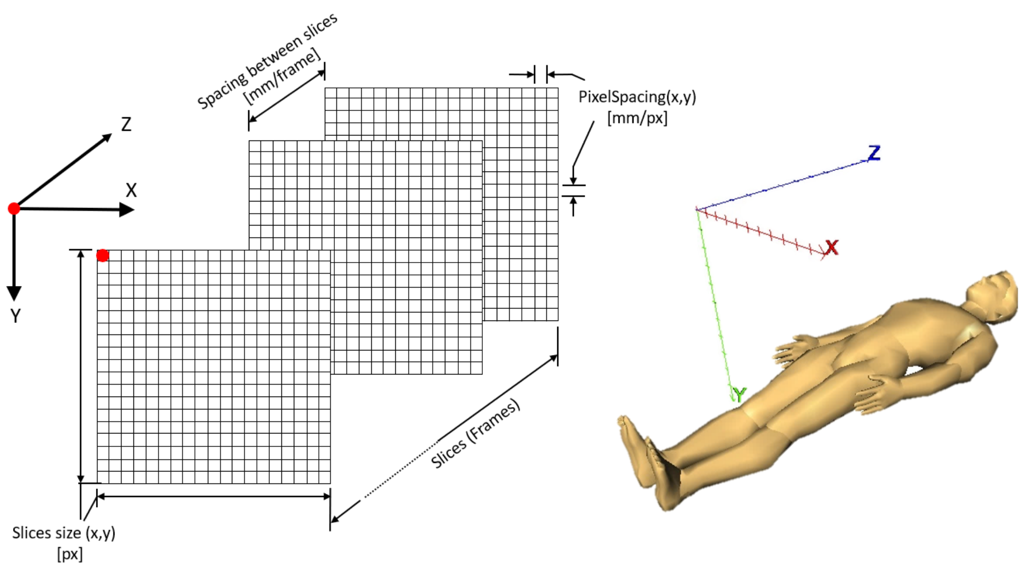

Figure A-8-1. DICOM structure

STL Format Structure

This CAD format represents only the 3D object surface using a list of facet data. Each facet is defined by three vertices and a normal unit vector. It is guaranteed by the format that each triangle shares two vertices with each of its adjacent triangles [199]. Figure A-8-2 shows the definition of a facet and interconnections among facets.

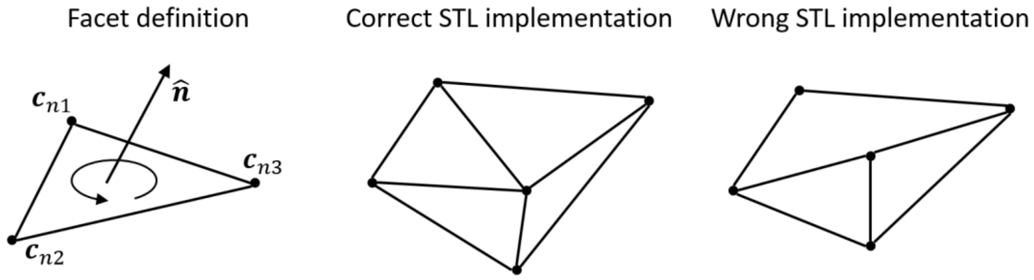

Figure A-8-2. STL structure

The vertex coordinates keep the selected dimension during the exporting time, usually millimeters, but other units can be found.

Conversion from STL structure to 3D Array

It is necessary to determine the 3D array size and initialize each voxel with a grayscale value of zero, meaning emptiness. As the STL file is created from CAD software; therefore, there is an involved coordinate frame in the CAD software that is kept during the exporting. The vertex list is analyzed. The minimum element of each coordinate is found and subtracted in each vertex, making each component of the vertex list equal or greater than zero. The size of the volume is found axis-wise as the maximum value in each coordinate seen in the entire list.

Once the 3D array is dimensioned, it must be filled using the facets data included in the STL file. Each facet can be seen as a triangle in 3D space. The sub-volume delimited by the facet in the 3D array is found. The lower and upper limits of the sub-volume are determined with the combination of maximum and minimum values per coordinate in the three vertices. Let $\boldsymbol{c}_{n1}$, $\boldsymbol{c}_{n2}$ and $\boldsymbol{c}_{n3} \in \mathbb{R}^3$ be the three vertices representing the facet n, $\Delta_n \in \mathbb{R}^3$, of the list. This sub-volume delimitation is done to minimize computation time in the coming steps; otherwise, the entire 3D array would be iterated to fill each facet. The sub-volume delimitation can be seen clearly in Figure A-8-3.

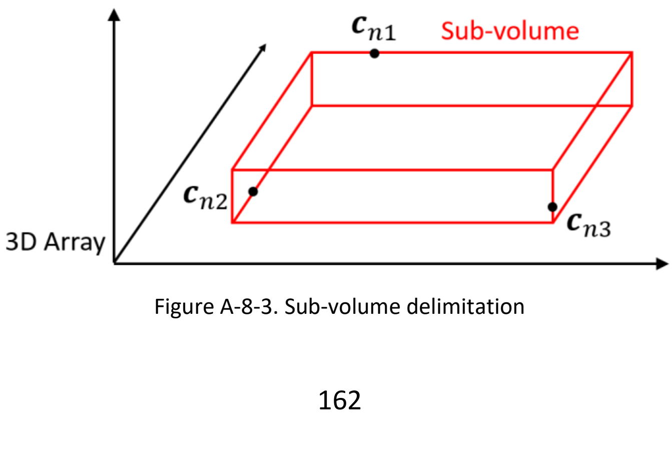

Figure A-8-3. Sub-volume delimitation

$\boldsymbol{c}_{n1}$, $\boldsymbol{c}_{n2}$ and $\boldsymbol{c}_{n3}$ represent a flat surface of the 3D object, so the entire facet must be filled. In other words, the voxels inside the triangle must have a gray-scale value different than zero. This gray-scale value is left as a user-input parameter. The filling process is carried out in three steps:

1 Let $S_n \in \mathbb{R}^3$ be the plane described by $\boldsymbol{c}_{n1}$, $\boldsymbol{c}_{n2}$ and $\boldsymbol{c}_{n3}$. In this step, the plane equation given $\boldsymbol{c}_{n1}$, $\boldsymbol{c}_{n2}$ and $\boldsymbol{c}_{n3}$ is calculated. That means finding the coefficients $a_{nn}, b_{nn}, c_{nn}, d_{nn} \in \mathbb{R}$ giving by the following the plane equation [195]:

$$a_{nn} \cdot x + b_{nn} \cdot y + c_{nn} \cdot z = d_{nn} \tag{172}$$

2 Iterate over every sub-volume voxel, and find the set of point that fulfills equation *(172)*.

Let $\boldsymbol{V}_n \in \mathbb{Z}^{p \times q \times r}$ be the sub-volume, which has to be filled. Every voxel of $\boldsymbol{V}_n$ is indexed by a point $\boldsymbol{c}_{pqr} = [x_p, y_q, z_r] \in \mathbb{Z}^3$. Substituting $\boldsymbol{c}_{pqr}$ in equation *(172)*

$$a_{nn} \cdot x_p + b_{nn} \cdot y_q + c_{nn} \cdot z_r - d_{nn} = \epsilon \tag{173}$$

In theory $\boldsymbol{c}_{pqr} \in S_n$ if $\epsilon = 0$, but due to the mix of discrete and real points, a small $\delta \in \mathbb{R}$ is allowed, such as $\boldsymbol{c}_{pqr}$ belongs to the plane if the absolute value of ϵ is smaller than δ.

$$\boldsymbol{c}_{pqr} \in S_n \text{ if } |\epsilon| < \delta \tag{174}$$

Using a trial-and-error approach, it is found that a $0.7 \leq \delta \leq 1.1$, leads to accurate representation of the CAD design as a 3D volume.

3 The set of points determined in the last step belong to the S_n, but it does not mean that they belong to Δ_n. It is necessary to further evaluate which of these points lie inside Δ_n. See Figure A-8-4.

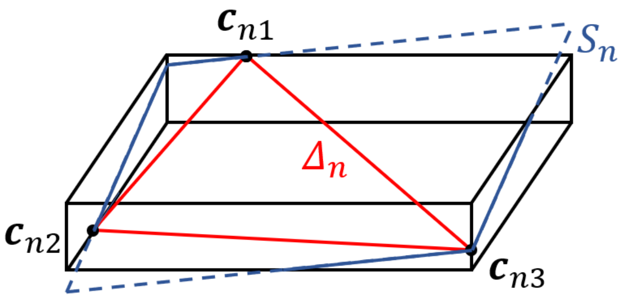

Figure A-8-4. Plane and facet inside sub-volume. Sub-volume in black, plane in blue and facet in red

Let $A_{n1n2n3} \in \mathbb{R}$ be the area of Δ_n, and let $\boldsymbol{c}_{nm} \in \mathbb{Z}^3$ be a point in the plane S_n. Let $A_{n1n2nm}, A_{n1nmn3}, A_{nmn2n3} \in \mathbb{R}$ be the areas of the triangles formed by the vertices $\boldsymbol{c}_{n1}$, $\boldsymbol{c}_{n2}$ and $\boldsymbol{c}_{nm}$, $\boldsymbol{c}_{n1}$, $\boldsymbol{c}_{nm}$ and $\boldsymbol{c}_{n3}$, and $\boldsymbol{c}_{nm}$, $\boldsymbol{c}_{n2}$ and $\boldsymbol{c}_{n3}$, respectively. $\boldsymbol{c}_{nm}$ belongs to the facet if the addition of A_{n1n2nm}, A_{n1nmn3} and A_{nmn2n3} is equal to A_{n1n2n3}. The working principle of determining points inside a triangle based on areas can be seen in Figure A-8-5.

$$\boldsymbol{c}_{nm} \in \Delta_n \text{ if } A_{n1n2nm} + A_{n1nmn3} + A_{nmn2n3} = A_{n1n2n3} \tag{175}$$

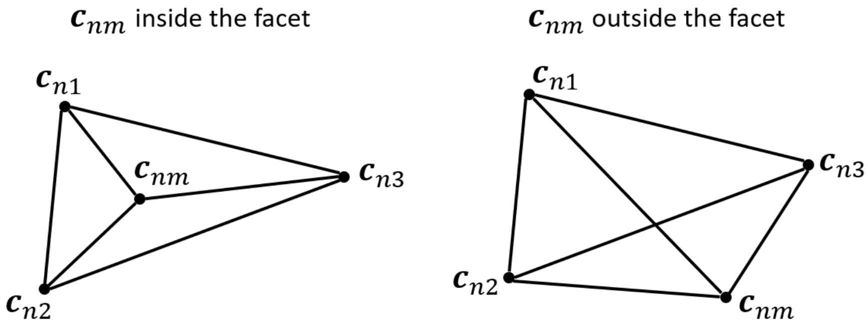

Figure A-8-5. Working principle of a point belonging to a facet based on triangles areas

Once again, the mix of integer and real points in the problem, makes the satisfaction of equation *(175)* unlikely. Let $\xi \in \mathbb{R}$ be a remaining error such as

$$A_{n1n2nm} + A_{n1nmn3} + A_{nmn2n3} - A_{n1n2n3} = \xi \tag{176}$$

$\boldsymbol{c}_{\mathrm{nm}}$ belongs to Δ_n if ξ is smaller than a define threshold. It is empirically found that a threshold value, $\zeta \in \mathbb{R}$, in the range from 9 to 17 facilitates the creation of facets. In summary from *(175)* and *(176)*

$$\boldsymbol{c}_{nm} \in \Delta_n \text{ if } \xi < \zeta \tag{177}$$

Conversion from 3D Array to DICOM

Each plane X-Y in the 3D array is then converted into an individual image. It means that the DICOM output is composed of many 2D Images (frames) as slices in the Z-component of the 3D array. Each image creation follows the DICOM format and is supported by the dicominfo and dicomwrite commands included in the MATLAB image processing toolbox. Due to the complexity of the DICOM format, the new DICOM is not made from scratch but by modifying an existing DICOM volume layout.

Appendix B. List of Publications

Alvarez-Gomez J, Jimenez G., Roth H, Wahrburg J.:" An Ultra-Fast Digitally Reconstructed Radiograph (DRR) Implementation of the Siddon-Jacobs Algorithm using Parallel Computing: Runtime Improvement of an Intensity-Based 2D/3D Registration," Current Directions in Biomedical Engineering 7(2), 25-28. DOI:10.1515/cdbme-2021-2007, Oct. 2021.

Alvarez-Gomez J, Roth H, Wahrburg J.:" Conception of a C-Arm Parametrization Device Evaluated by an Inverse Registration Approach: An application for 2D/3D Registration in Minimally Invasive Surgery," Current Directions in Biomedical Engineering 7(1), 26-29. DOI:10.1515/cdbme-2021-1006, Aug. 2021.

Alvarez-Gomez J, Roth H, Wahrburg J.:" Intra-Operative Registration of 2D C-Arm Images with Pre-Operative 3D CT-Scan Data in Computer Assisted Spine Surgery," Industrial and Medical Measurement and Sensor Technology Vehicle Sensor Technology. IEEE Workshop Sensorica 2020, 38-39. ISBN: 978-3-946757-03-0. May, 2020.

Alvarez-Gomez, J.; Spieß, A.; Roth, H.; Wahrburg, J.: "A Deep-Learning Approach to Detect Fiducials in Planar X-Ray Images for Undistortion of Conventional C-Arm Images," Current Directions in Biomedical Engineering 6(3),40-43. DOI:10.1515/cdbme-2020-3011, Nov. 2020.

Alvarez-Gomez J, Roth H, Wahrburg J.:" Intraoperative Registration of 2D C-arm Images with Preoperative CT Data in Computer Assisted Spine Surgery: Motivation to Use Convolutional Neural Networks for Initial Pose Generator." Current Directions in Biomedical Engineering 5(1): 25-28. DOI:10.1515/cdbme-2019-0007, Sep. 2019.

Alvarez-Gomez J, Roth H, Wahrburg J.:" Comparison of Similarity Measurements and Optimizers for Intraoperative Registration of 2D C-arm Images with Preoperative CT Data in Computer-Assisted Spine Surgery." CURAC 2019 (18th Annual Meeting of the German Society for Computer- and Robot-Assisted Surgery). Volume: ISBN: 978-3-00-063717-9, Sep. 2019.

Conferences

Alvarez-Gomez, J.; Spieß, A.; Roth, H.; Wahrburg, J.: "A Deep-Learning Approach to Detect Fiducials in Planar X-Ray Images for Undistortion of Conventional C-Arm Images. " BMT 2020, 29.09.-01.10.2020. Leipzig, Germany (Online).

Alvarez-Gomez J, Roth H, Wahrburg J.: "Intraoperative Registration of 2D C-arm Images with Preoperative CT Data in Computer Assisted Spine Surgery: Motivation to Use Convolutional Neural Networks for Initial Pose Generator. " BMT 2019, 25.-26.09.2019. Frankfurt am Main, Germany.

Alvarez-Gomez J, Roth H, Wahrburg J.: "Comparison of Similarity Measurements and Optimizers for Intraoperative Registration of 2D C-arm Images with Preoperative CT Data in Computer-Assisted Spine Surgery." CURAC 2019, 19.-21.09.2019. Reutlingen, Germany.

Alvarez-Gomez J, Roth H, Wahrburg J.: "Intraoperative Registration of 2D C-Arm Images with Preoperative CT Data in Computer Assisted Spine Surgery". NSpine Main Conference 2019, 01.-03.07.2019. London, UK.

Appendix C. References

[1] B. Staub and S. Sadrameli, "The use of robotics in minimally invasive spine surgery," *Journal of Spine Surgery,* vol. 5, no. Suppl 1, pp. 31-40, 2019.

[2] C. Vo, B. Jiang, T. Azad, N. Crawford, A. Bydon and N. Theodore, "Robotic Spine Surgery: Current State in Minimally Invasive Surgery," *Global Spine Journal,* vol. 10, no. suppl 2, pp. 34S-40S, 2020.

[3] A. Vaishnav, Y. Othman, S. Virk, C. Gang and S. Qureshi, "Current state of minimally invasive spine surgery," *Journal of Spine Surgery,* vol. 5, no. suppl 1, pp. S2-S10, 2019.

[4] J. Palep, "Robotic assisted minimally invasive surgery," *Journal of Minimal Access Surgery,* vol. 5, no. 1, pp. 1-7, 2009.

[5] S. Aouadi and L. Sarry, "Accurate and precise 2D–3D registration based on X-ray intensity," *Computer Vision and Image Understanding,* vol. 110, no. 1, pp. 134-151, 2008.

[6] Y. Otake, A. Wang, W. S., A. Uneri, G. Kleinszig, S. Vogt, A. Khanna, Z. Gokaslan and J. Siewerdsen, "Robust 3D-2D image registration: application to spine interventions and vertebral labeling in the presence of anatomical deformation," *Physics in Medicine and Biology,* vol. 58, no. 23, p. 8535–8553, 2013.

[7] I. Marintschev, F. Gras, K. Klos, A. Wilharm, T. Mückley and G. Hofmann, "Navigation of vertebro-pelvic fixations based on CT-fluoro matching," *European spine journal : official publication of the European Spine Society, the European Spinal Deformity Society, and the European Section of the Cervical Spine Research Society,* vol. 19, no. 11, p. 1921–1927, 2010.

[8] P. Knappe, "Konzeption und Realisierung eines navigierten interaktiven Chirurgierobotersystems," Dissertation, Universität Siegen, Siegen, 2007.

[9] S. Gertzbein and S. Robbins, "Accuracy of Pedicular Screw Placement in Vivo," *Spine,* vol. 15, no. 1, pp. 11-14, 1990.

[10] R. Mobbs and D. Raley, "Complications With K-Wire Insertion for Percutaneous Pedicle Screws," *Journal of Spinal Disorders and Techniques,* vol. 27, no. 7, pp. 390-394, 2014.

[11] J. Wahrburg, "Kapitel 1: Kinematikmodell," in *Skript zur Vorlesung Robotik 1*, Siegen, University of Siegen, 2010, pp. 2-11.

[12] J. Diebel, "Representing Attitude: Euler Angles, Unit Quaternions, and Rotation Vectors," 2006.

[13] S. Hollasch, "Avanced animation and rendering techniques," *Computers & Graphics,* vol. 18, no. 2, p. 249, 1994.

[14] P. Singla, D. Mortari and J. Junkins, "How to Avoid Singularity For Euler Angle Set?," in *Proceedings of the AAS Space Flight Mechanics Conference*, Hawaii, 2004.

[15] O. Gieseler, J. Alvarez-Gomez, H. Roth and J. Wahrburg, "Design of smart tools to support pre- and intra-operative use of surgical navigation systems," *Technisches Messen,* vol. 85, no. 5, pp. 351-358, 2018.

[16] S. Sahm, H. Roth, D. Scale, A. Von Schillling, G. Winkler, U. Spetzger and J. Wahrburg, "An approach for minimal invasive insertion of pedicle screws using a navigated robotic assistance system," in *14th Annual Meeting of CAOS-International*, Milan, Italy, 2014.

[17] A. Samdani, A. Ranade, D. Sciubba, P. Cahill, M. Antonacci, D. Clements and R. Betz, "Accuracy of free-hand placement of thoracic pedicle screws in adolescent idiopathic scoliosis: how much of a difference does surgeon experience make?," *European spine journal,* vol. 19, no. 1, pp. 91-95, 2010.

[18] B. Jiang, K. A., C. Zygourakis, S. Kalb, A. Zhu, J. Godzik, C. Molina, A. Blitz, A. Bydon, N. Crawford and N. Theodore, "Pedicle screw accuracy assessment in ExcelsiusGPS® robotic spine surgery: evaluation of deviation from pre-planned trajectory," *Chinese neurosurgical journal,* vol. 4, p. 23, 2018.

[19] D. Devito, L. Kaplan, R. Dietl, M. Pfeiffer, D. Horne, B. Silberstein, M. Hardenbrook, G. Kiriyanthan, Y. Barzilay, A. Bruskin, D. Sackerer, V. Alexandrovsky, C. Stüer, R. Burger, J. Maeurer, G. Donald, D. Gordon and R. Schoenmayr, "Clinical acceptance and accuracy assessment of spinal implants guided with SpineAssist surgical robot: retrospective study," *Spine,* vol. 35, no. 24, p. 2109–2115, 35.

[20] Y. Fan, J. Peng Du, J. Liu, J. Zhang, S. Liu and D. Hao, "Radiological and clinical differences among three assisted technologies in pedicle screw fixation of adult degenerative scoliosis," *Scientific reports,* vol. 8, no. 1, p. 890, 2018.

[21] O. Bott, M. Wagner, C. Duwenkamp, N. Hellrung and K. Dresing, "Improving education on C-arm operation and radiation protection with a computer-based training and simulation system," *Int J CARS,* vol. 4, pp. 399-407, 2009.

[22] L. H. Wang, A. M. McKenzie-Brown and A. H. Hord, in *Handbook of C-Arm Fluroscopy-Guided Spinal Injections*, CRC Press, 2006, pp. 7-13.

[23] Ziehm imaging GmbH, "Ziehm Vario 3D - Technical Manual," Ziehm Vario, Nuremberg, 2006.

[24] M. Lavine, "The Early Clinical X-Ray in the United States: Patient Experiences and Public Perceptions," *Journal of the History of Medizine and Allied Sciences,* vol. 67, no. 8, pp. 587-625, 2011.

[25] B. Schueler, "The AAPM/RSNA Physics Tutorial for Residents. General Overview of Fluoroscopic Imaging," *RadioGraphics,* vol. 20, no. 4, pp. 1115-1126, 2000.

[26] A. Kedgley, A. Fox and R. Jenkyn, "Image intensifier distortion correction for fluoroscopic RSA: the need for independent accuracy assessment," *Journal of applied clinical medical physics,* vol. 13, no. 1, pp. 197-204, 2012.

[27] T. Abbas, "Correction of Distortion Aberration in Electron Magnetic Lenses," *International Journal of Applied Physics,* vol. 3, no. 2, pp. 1-5, 2016.

[28] A. Hebecker, "C-Arm-Based Navigation," in *Navigation and Robotics in Total Joint and Spine Surgery*, Berlin, Heidelberg, New York, Springer Science & Business Media, 2012, pp. 17-23.

[29] A. Jain, A. Deguet, I. Iordachita, G. Chintalapani, J. Blevins, Y. Le, E. Armour, C. Burdette, D. Song and G. Fichtinger, "Intraoperative-3D Guidance in Prostate Brachytherapy Using a Non-isocentric C-arm," *Lectures Notes in Computer Science,* vol. 4792, no. 2, pp. 9-17, 2007.

[30] E. Nickoloff, "AAPM/RSNA Physics Tutorial for Residents: Physics of Flat-Panel Fluoroscopy Systems. Survey of Modern Fluoroscopy Imaging: Flat-Panel Detectors versus Image Intensifiers and More," *RadioGraphics,* vol. 31, no. 2, pp. 591-602, 2011.

[31] R. Hartley and A. Zisserman, Multiple View Geometry in Computer Vision, 2nd ed., New York: Cambridge University Press, 2004.

[32] C. Mennessier, B. Spencer, R. Clackdoyle, A.-C. Conneau and T. Xu, "Distortion Correction, Geometric Calibration, and Volume Reconstruction for an Isocentric C-Arm X-Ray System," *IEEE Nuclear Science Symposium Conference Record,* pp. 2943-2947, 2011.

[33] Z. Yaniv, L. Joskowicz, A. Simkin, M. Garza-Jinich and C. Milgrom, "Fluroscopic Image Processing for Computer-Aided Ortohpaedic Surgery," *International Conference on Medical Image Computing and Computer-Assisted Intervention,* vol. 1496, pp. 325-334, 1998.

[34] H. Gernsheim and A. Gernsheim, "Re-Discovery of the World's First Photograph," *The Photographic Journal,* pp. 118-120, 1952.

[35] J. Davidson and K. Hunt, Robots and Screw Theory: Applications of Kinematics and Statics to Robotics, New York: Oxford University Press, 2004.

[36] C. Mennessier, R. Clackdoyle and F. Noo, "Direct determination of geometric alignment parameters for cone-beam scanners," *Physics in Medicine and Biology,* vol. 54, no. 6, pp. 1633-1660, 2009.

[37] H. Ha, S. Jeon, S. Lee and e. al., "Perspective pinhole model with planar source for augmented reality surgical navigation based on C-arm imaging," *Int J CARS,* vol. 13, pp. 1671-1682, 2018.

[38] S. Thürauf, O. Hornung, M. Körner, F. Vogt, A. Knoll and M. Nasseri, "Model-Based Calibration of a Robotic C-Arm System Using X-Ray Imaging," *Journal of Medical Robotics Research,* vol. 3, no. 3, p. 12, 2018.

[39] A. Akritas, G. Malaschonok and P. Vigklas, "The SVD-Fundamental Theorem of Linear Algebra," *Nonlinear Analysis: Modelling and Control,* vol. 11, no. 2, pp. 123-136, 2006.

[40] O. Faugeras, Three-Dimensional Computer Vision A Geometrical Viewpoint, Cambridge, Mass: The MIT Press, 1993.

[41] B. Bellekens, V. Spruyt, R. Berkvens and M. Weyn, "A Survey of Rigid 3D Pointcloud Registration Algorithms," in *AMBIENT 2014 : The Fourth International Conference on Ambient Computing, Applications, Services and Technologies*, Rome, 2014.

[42] P. Van den Elsen, E. Pol and M. Viergever, "Medical image matching-a review with classification," *IEEE Engineering in Medicine and Biology Magazine,* vol. 12, no. 1, pp. 26-39, 1993.

[43] J. Maintz and M. Viergever, "A Survey of Medical Image Registration," *Medical Image Analysis,* vol. 2, no. 1, p. 1–37, 1998.

[44] Z. Li, K. Li, K. Chen, S. Liang, J. Gu and L. Wang, "Comparison of 2D and 3D Ultrasound Guided Percutaneous Renal Puncture," *IFMBE Proceedings,* vol. 39, pp. 708-711, 2013.

[45] J. Qiao, F. Zhu, L. Xu, Z. Liu, X. Sun, B. Qian, Q. Jiang, Z. Zhu and Y. Qiu, "Accuracy of pedicle screw placement in patients with Marfan syndrome," *BMC Musculoskeletal Disorders,* vol. 18, p. 123, 2017.

[46] P. Katonis, J. Christoforakis, A. Aligizakis, C. Papadopoulos, G. Sapkas and A. Hadjipavlou, "Complications and Problems Related to Pedicle Screw Fixation of the Spine," *Clinical Orthopaedics and Related Research,* vol. 411, pp. 86-94, 2003.

[47] V. Kuperman, "MRI Instrumentation: Magnets, Gradient, Coils, and Radiofrequency Coils," in *Magnetic Resonance Imaging. Physical Principles and Applications*, San Diego, Academic Press, 2000, pp. 1-8.

[48] V. Kuperman, "MRI Instrumentation: Magnets, Gradient, Coils, and Radiofrequency Coils," in *Magnetic Resonance Imaging: Physical Principles and Applications*, San Diego, Academic Press, 2000, pp. 151-171.

[49] S. A. Huettel, "An Introduction to fMRI," in *Functional Magnetic Resonance Imaging*, 2nd ed., Sinauer Associates, Inc, 2009, pp. 3-4.

[50] R. Ramlau and O. Scherzer, "The first 100 years of the Radon transform," *Inverse Problems,* vol. 34, no. 9, pp. E1-E4, 2018.

[51] J. Hsieh, "Computed Tomography: Principles, Desing, Artifacts, and Recent Advances," SPIE and Wiley-Interscience, 2009, p. 207.

[52] X. L. S. Li, "A compressed sensing-based iterative algorithm for CT reconstruction and its possible application to phase contrast imaging," *BioMed Eng OnLine,* vol. 10, no. 73, pp. 1-14, 2011.

[53] N. J. Pelc, "Recent and Future Directions in CT Imaging," *Annals of Biomedical Engineering,* vol. 42, no. 2, pp. 260-268, 2014.

[54] S. Eun, H. Lee, S. Lee, K. Kim and W. Liu, "MRI versus CT for the diagnosis of lumbar spinal stenosis," *Journal of Neuroradiology,* vol. 39, no. 2, pp. 104-109, 2012.

[55] L. Tan, M. Kasliwal and V. Traynelis, "Comparison of CT and MRI findings for cervical spine clearance in obtunded patients without high impact trauma," *Clinical Neurology and Neurosurgery,* vol. 120, pp. 23-26, 2014.

[56] N. Hosten and T. Liebig, CT of the Head and Spine, Thieme: Verlag, 2002, p. 282.

[57] G. Penney, J. Weese, J. Little, P. Desmedt, D. Hill and D. Hawkes, "A Comparison of Similarity Measures for Use in 2-D–3-D Medical Image Registration," *IEEE Transactions on Medical Imaging,* vol. 17, no. 4, pp. 586-595, 1998.

[58] P. Markelj, D. Tomaževič, B. Likar and F. Pernuš, "A review of 3D/2D registration methods for image-guided interventions," *Medical Image Analysis,* vol. 16, no. 3, pp. 642-661, 2012.

[59] T. De Silva, A. Uneri, M. Ketcha, S. Reaungamornrat, J. Goerres, M. Jacobson, S. Vogt, G. Kleinszig, A. Khanna, J. Wolinsky and J. Siewerdsen, "Registration of MRI to intraoperative radiographs for target localization in spinal interventions," *Physics in Medicine & Biology,* vol. 62, no. 2, pp. 684-701, 2017.

[60] M. Van der Bom, J. Pluim, M. Gounis, E. van de Kraats, S. Sprinkhuizen, J. Timmer, R. Homan and L. Bartels, "Registration of 2D x-ray images to 3D MRI by generating pseudo-CT data," *Physics in Medicine and Biology,* vol. 56, no. 4, pp. 1031-1043, 2011.

[61] I. Hacihaliloglu, "Ultrasound imaging and segmentation of bone surfaces: A review," *TECHNOLOGY,* vol. 5, no. 2, pp. 74-80, 2017.

[62] O. S. Pianykh, "Chapter 6: Medical Images in DICOM," in *Digital Imaging and Communications in Medicine (DICOM)*, Boston, Springer, 2008, pp. 104-405.

[63] NEMA, "Table C.7-10. Image Plane Module Attributes," National Electrical Manufacturers Association, [Online]. Available: http://dicom.nema.org/medical/dicom/current/output/html/part03.html#table_C.7-10. [Accessed 06. 11. 2020].

[64] NEMA, "C.7.6.2.1.1 Image Position and Image Orientation," National Electrical Manufacturers Association, [Online]. Available: http://dicom.nema.org/medical/dicom/current/output/html/part03.html#sect_C.7.6.2.1.1. [Accessed 06. 11. 2020].

[65] G. Tam, Z. Cheng, Y. Lai, F. Langbein and Y. Liu, "Registration of 3D Point Clouds and Meshes:," *IEEE Transactions on Visualization and Computer Graphics,* vol. 19, no. 7, pp. 1199-1217, 2013.

[66] J. Liu, G. Singh, S. Al'Aref, B. Lee, O. Oleru and J. Min, "Image Registration in Medical Robotics and Intelligent Systems: Fundamentals and Applications," *Advanced Intelligent Systems.Adv. Intell. Syst., 1: 1900048.,* pp. 1-16, 2019.

[67] D. Hawkes, L. Robinson, J. Crossman, H. Sayman, R. Mistry and e. al., "Registration and display of the combined bone scan and radiograph in the diagnosis and management of wrist injuries," *European Journal of Nuclear Medicine,* vol. 18, pp. 752-756, 1991.

[68] H. Keil, N. Beisemann, B. Swartman, S. Vetter, P. Grützner and J. Franke, "Intra-operative imaging in trauma surgery," *EFORT Open Reviews,* vol. 3, pp. 541-549, 2018.

[69] J. Yao and R. Taylor, "Assessing Accuracy Factors in Deformable 2D/3D Medical Image Registration Using a Statistical Pelvis Model," *Proceedings of the Ninth IEEE International Conference on Computer Vision,* vol. 2, pp. 1329-1334, 2003.

[70] G. Zheng, X. Zhang and L. Nolte, "Assessing Spline-Based Multi-resolution 2D-3D Image Registration for Practical Use in Surgical Guidance," *Lecture Notes in Computer Science,* vol. 3150, pp. 294-301, 2004.

[71] F. Alam, S. Rahman, S. Ullah, A. Khalil and A. Din, "A review on extrinsic registration methods for medical images," *Technical Journal, University of Engineering and Technology,* vol. 21, no. 3, pp. 110-119, 2016.

[72] D. Winkler, H. Vitzthum and S. H., "Spinal Markers: a New Method for Increasing," *Computer Aided Surgery,* vol. 4, no. 2, pp. 101-104, 1999.

[73] B. Reto, "Multimodality Registration in Daily Clinical Practice," in *Mathematical Models for Registration and Applications to Medical Imaging*, O. Scherzer, Ed., Springer, 2005, pp. 165-183.

[74] X. Wang, E. S., F. M., S. S. and F. D., "Data Registration and Fusion," in *Biomecial Information Technology*, F. D., Ed., Academic Press, 2008, pp. 187-210.

[75] M. Watson and R. Maciunas, "Frameless Stereotactic Systems: General Considerations," in *Handbook of Stereotactic and Functional Neurosurgery*, M. Schulder, Ed., Marcel Dekker, Inc, 2003, pp. 55-63.

[76] L. Zamorano, F. Vinas, Z. Jiang and F. Diaz, "Use of Surgical Wands in Neurosurgery," in *Advances and Technical Standards in Neurosurgery*, vol. 24, F. Cohadon, Ed., Vienna, Springer, 1998, pp. 77-128.

[77] O. Wieben, "Image-Guided Surgery," in *Minimally Invasive Medical Technology*, J. Webster, Ed., Boca Raton, CRC Press, 2016, p. 24.

[78] J. Maintz and M. Viergever, "An Overview of Medical Image Registration Methods," *Medical Image Analysis,* vol. 2, no. 1, pp. 1-36, 1998.

[79] A. Farag, S. Yamany, J. Nett, T. Moriarty, A. El-Baz, S. Hushek and R. Falk, "Medical Image Registration: Theory, Algorithm, and Case Studies in Surgical Simulation, Chest Cancer, and Multiple Sclerosis," in *Handbook of Biomedical Image Analysis: Registration Models*, New York, Boston, Dordrecht, London, Moscow, Kluwe Academic/ Plenum Publishers, 2005, pp. 1-47.

[80] X. Dai and S. Khorram, "A Feature-Based Image Registration Algorithm Using Improved Chain-Code Representation Combined with Invariant Moments," *IEEE Transactions on Geoscience and Remote Sensing,* vol. 37, no. 5, pp. 2351-2362, 1999.

[81] C. Cyr, A. Kamal, T. Sebastian and B. Kimia, "2D-3D Registration Based on Shape Matching," *Proceedings of the IEEE Workshop on Mathematical Methods in Biomedical Image Analysis,* pp. 198-203, 2000.

[82] H. Mahmoud, F. Masulli and S. Rovetta, "Feature-Based Medical Image Registration Using a Fuzzy Clustering Segmentation Approach," *Lecture Notes in Computer Science,* vol. 7845, pp. 37-47, 2013.

[83] B. Fei, J. Suri and D. Wilson, "Three-Dimensional Rigid and Non-Rigid Image Registration for the Pelvis and Prostate," in *Handbook of Biomedical Image Analysis: Registration Models*, New York, Boston, Dordrecht, London, Moscow, Kluwer Academic / Plenum Publishers, 2005, pp. 103-150.

[84] T. D. Alter and D. Jacobs, "Error propagation in 3D-from-2D recognition: Scaled-orthographic and perspective projections," *Proceedings: ARPA Image Understanding Workshop, Monterey, California,* vol. 2, pp. 973-980, 1994.

[85] V. Gandhi and S. Panchal, "Feature Based Image Registration Techniques: An Introductory Survey," *International Journal of Engineering Development and Research,* vol. 2, no. 1, pp. 368-375, 2014.

[86] B. Islam and J. Kabir, "A New Feature-Based Image Registration Algorithm," *Computer Technology and Application,* vol. 4, pp. 79-84, 2013.

[87] E. B. Van de Kraats, G. P. Penney, T. van Walsum and W. J. Niessen, "Multispectral MR to X-Ray Registration of Vertebral Bodies by Generating CT-Like Data," *Medical Image Computing and Computer-Assisted Intervention,* pp. 911-918, 2005.

[88] Z. Zhong, X. Guo, Y. Cai, Y. Yang, J. Wang, X. Jia and W. Mao, "3D-2D Deformable Image Registration Using Feature-Based Nonuniform Meshes," *BioMed Research International,* vol. 2016, pp. 1-19, 2016.

[89] J. Wang, W. Zhang and M. Chen, "A Survey of 3D Image Navigation and High Precision Dynamic Registration in Minimally Invasive Surgery," *Procedia Computer Science,* vol. 131, p. 320–326, 2018.

[90] R. Mclauglin, J. Hipwell, G. Penney, K. Rhode, A. Chung, J. Noble and D. Hawkes, "Intensity-based Registration versus Feature-based registration for neurointerventions," *Proceedings of Medical Image Understanding and Analysis,* pp. 69-72, 2001-2002.

[91] R. McLaughlin, J. Hipwell, D. Hawkes, J. Noble, J. Byrne and T. Cox, "A Comparison of 2D-3D Intensity-Based Registration and Feature-Based Registration for Neurointerventions," *Lecture Notes in Computer Science,* vol. 2489, pp. 517-524, 2002.

[92] D. Tomazevic, B. Likar, T. Slivnik and F. Pernus, "3-D/2-D Registration of CT and MR to X-Ray Images," *IEEE Transactions on Medical Imaging,* vol. 22, no. 11, pp. 1407-1416, 2003.

[93] H. Livyatan, Z. Yaniv and L. Joskowicz, "Gradient-based 2-D/3-D rigid registration of fluoroscopic X-ray to CT," *IEEE Transactions on Medical Imaging,* vol. 22, no. 11, pp. 1395-1406, 2003.

[94] P. Markelj, D. Tomazevic, P. F. and B. Likar, "Robust Gradient-Based 3-D/2-D Registration of CT and MR to X-Ray Images," *IEEE Transactions on Medical Imaging,* vol. 27, no. 12, pp. 1704-1714, 2008.

[95] A. Schweikard, H. Shiomi and J. Adler, "Respiration tracking in radiosurgery without fiducials," *Medical Robotics and Computer Assisted Surgery,* vol. 1, no. 2, pp. 19-27, 2005.

[96] D. Knaan and L. Joskowicz, "Effective Intensity-Based 2D/3D Rigid Registration between Fluoroscopic X-Ray and CT," *Medical Image Computing and Computer-Assisted Intervention - MICCAI 2003,* vol. 2878, pp. 351-358, 2003.

[97] D. Russakoff, T. Rohlfing, A. Ho, D. Kim, R. Shahidi, J. Adler and C. Maurer, "Evaluation of Intensity-Based 2D-3D Spine Image Registration Using Clinical Gold-Standard Data," *Biomedical Image Registration. WBIR 2003,* vol. 2717, pp. 151-160, 2003.

[98] C. Gendrin, P. Markelj, S. Pawiro, J. Spoerk, C. Bloch, C. Weber, M. Figl, H. Bergmann, W. Birkfellner, B. Likar and F. Pernuš, "Validation for 2D/3D registration II: The comparison of intensityand gradient-based merit functions using a new gold standard data set," *Medical Physics,* vol. 38, no. 3, pp. 1491-1502, 2011.

[99] W. Wein, "Intensity Based Rigid 2D-3D Registration Algorithms for Radiation Therapy," Diplomarbeit, Technische Universität München, Munich, 2003.

[100] D. Russakoff and a. et, "Fast Generation of Digitally Reconstructed Radiographs Using Attenuation Fields With Application to 2D-3D Image Registration," *IEEE Transactions on Medical Imaging,* vol. 24, no. 11, pp. 1441-1454, 2005.

[101] D. Weiskopf, GPU-Based Interactive Visualization, Berlin, Heidelberg: Springer, 2007.

[102] S. Roettger, D. Weiskopf, T. Ertl and W. Strasser, "Smart Hardware-Accelerated Volume Rendering," *Proceedings of the symposium on Data visualisation 2003 (VISSYM '03),* pp. 231-238, 2003.

[103] R. Siddon, "Fast calculation of the exact radiological path for a three-dimensional CT array," *Medical Physics,* vol. 12, no. 2, pp. 252-255, 1985.

[104] 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,* vol. 6, pp. 89-94, 1998.

[105] M. Pappalardo, G. Mastroeni and M. Passacantando, "Merit functions: A bridge between optimization and equilibria," *4OR quarterly journal of the Belgian, French and Italian Operations Research Societies,* vol. 12, no. 1, pp. 1-33, 2014.

[106] D. Maravall and M. Patricio, "Image Segmentation and Pattern Recognition," in *Pattern Recognition and String Matching*, Dordrecht, Boston, London, Kluwer academic publishers, 2002, pp. 411-464.

[107] J. Fitzpatrick, D. Hill and C. Maurer, "Image Registration," in *Handbook of Medical Imaging: Medical Image Processing and Analysis*, Bellingham, Washington, SPIE Press, 2004, pp. 447-514.

[108] D. Hill, M. Batchelor, M. Holden and D. Hawkes, "Medical Image Registration," *Physics in Medicine and Biology,* vol. 46, pp. R1-R45, 2001.

[109] A. Goshtasby, Theory and Applications of Image Registration, John Wiley & Sons, Inc., 2017.

[110] H. Niitsuma and T. Maruyama, "Sum of Absolute Difference Implementations for Image Processing on FPGAs," *International Conference on Field Programmable Logic and Applications,* pp. 167-170, 2010.

[111] W. Birkfellner, M. Stock, M. Figl, C. Gendrin, J. Hummel, S. Dong, J. Kettenbach, D. Georg and H. Bergmann, "Stochastic rank correlation: A robust merit function for 2D/3D registration of image data obtained at different energies," *Medical Physics,* vol. 36, no. 8, pp. 3420-3428, 2010.

[112] P. Bifulco, M. Cesarelli, R. Allen, M. Sansone and M. Bracale, "Automatic recognition of vertebral landmarks in fluoroscopic sequences for analysis of intervertebral kinematics," *Medical & Biological Engineering & Computing,* vol. 39, no. 1, pp. 65-75, 2001.

[113] C. Bradley, "The Absolute Correlation Coefficient," *The Mathematical Gazette,* vol. 69, no. 447, pp. 12-17, 1985.

[114] J. Wu, M. Kim, J. Peters, H. Chung and S. Samant, "Evaluation of similarity measures for use in the intensity-based rigid 2D-3D registration for patient positioning in radiotherapy," *Medical Physics,* vol. 36, no. 12, p. 5391–5403, 2009.

[115] J. Weese, T. Buzug, C. Lorenz and C. Fassnacht, "An Approach to 2D/3D Registration of a Vertebra in 2D X-ray Fluoroscopies with 3D CT Images," *Lecture Notes in Computer Science,* vol. 1205, pp. 119-128, 1997.

[116] J. Kim, S. Li, D. Pradhan, R. Hammoud, Q. Chen, F. Yin, Y. Zhao, J. Kim and B. Movsas, "Comparison of similarity measures for rigid-body CT/Dual X-ray image registrations," *Technology in Cancer Research and Treatment,* vol. 6, no. 4, pp. 337-345, 2007.

[117] I. Sobel and G. Feldman, "A 3x3 Isotropic Gradient Operator for Image Processing," in *Stanford Artificial Intelligence Project (SAIL)*, Stanford, 1968.

[118] I. Van der Bom, S. Klein, M. Staring, R. Homan, L. Bartels and J. Pluim, "Evaluation of optimization methods for intensity-based 2D-3D registration in x-ray guided interventions," *Medical Imaging 2011: Image Processing,* vol. 7962, p. 15, 2011.

[119] D. LaRose, "Iterative X-ray/CT Registration Using Accelerated Volume Rendering," PhD thesis, Robotics Institute, Carnegie Mellon University, Pittsburgh, PA, 2001.

[120] D. Van der Weken, M. Nachtegael and E. Kerre, "Using Similarity Measures for Histogram Comparison," *Lecture Notes in Computer Science,* vol. 2715, pp. 396-403, 2003.

[121] D. Hill, "Across-Modality Registration Using Intensity-Based Cost Functions," in *Handbook of Medical Image Processing and analysis*, Amsterdam, Academic Press, 2009, pp. 613-628.

[122] P. Viola and W. Wells, "Alignment by Maximization of Mutual Information," *Proceedings of IEEE International Conference on Computer Vision,* pp. 16-23, 1995.

[123] L. Zollei, E. Grimson, A. Norbash and W. Wells, "2D-3D rigid registration of X-ray fluoroscopy and CT images using mutual information and sparsely sampled histogram estimators," *Proceedings of the 2001 IEEE Computer Society Conference on Computer Vision and Pattern Recognition,* vol. 2, pp. II-696-II-703, 2001.

[124] J. Pluim, J. Maintz and M. Viergever, "Mutual-information-based registration of medical images: a survey," *IEEE Transactions on Medical Imaging,* vol. 22, no. 8, pp. 986-1004, 2003.

[125] T. Kvålseth, "On Normalized Mutual Information: Measure Derivations and Properties," *Entropy,* vol. 19, no. 11, p. 631, 2017.

[126] A. Roche, G. Malandain, X. Pennec and N. Ayache, "Multimodal Image Registration by Maximization of the Correlation Ratio," RR-3378, INRIA. 1998. ffinria-00073311, 1998.

[127] A. Roche, G. Malandain, X. Pennec and N. Ayache, "The correlation ratio as a new similarity measure for multimodal image registration," *Lecture Notes in Computer Science,* vol. 1496, pp. 115-1124, 1998.

[128] M. Brown, D. Windridge and J. Guillemaut, "A Family of Globally Optimal Branch-and-Bound Algorithms for 2D-3D Correspondence-Free Registration," *Pattern Recognition,* vol. 93, pp. 36-54, 2019.

[129] L. Lemieux, R. Jagoe, D. Fish, N. Kitchen and D. Thomas, "A patient-to-computed-tomography image registration method based on digitally reconstructed radiographs," *Medical Physics,* vol. 21, no. 11, pp. 1749-1760, 1994.

[130] K. Gilhuijs, P. van de Ven and M. van Herk, "Automatic threedimensional inspection of patient setup in radiation therapy using portal images, simulator images, and computed tomography data," *Medical Physics,* vol. 23, no. 3, pp. 389-399, 1996.

[131] A. Sayed, M. Sardeshmukh and S. Limkar, "Optimisation Using Levenberg-Marquardt Algorithm of Neural Networks for Iris," *Advances in Intelligent Systems and Computing,* vol. 247, p. 91–98, 2014.

[132] W. Press, S. Teukolsky, W. Vetterling and B. Flannery, Numerical Recipes, Third ed., Cambridge: Cambridge University Press, 2007.

[133] J. Nelder and R. Mead, "A Simplex Method for Function Minimization," *The Computer Journal,* vol. 7, no. 4, p. 308–313, 1965.

[134] A. Cole-Rhodes and R. Eastman, "Gradient descent approaches to image registration," in *Image Registration for Remote Sensing*, Cambridge, Cambridge University Press, 2011, p. 265–275.

[135] J. Duchi, E. Hazan and Y. Singer, "Adaptive Subgradient Methods for Online Learning and Stochastic Optimization," *Journal of Machine Learning Research,* vol. 12, no. 61, pp. 2121-2159, 2011.

[136] M. Zeiler, "ADADELTA: An Adaptive Learning Rate Method," *ArXiv,* vol. abs/1212.5701, 2012.

[137] D. Kingma and J. Ba, "Adam: a Method for Stochastic Optimization," *International Conference on Learning Representations,* pp. 1-15, 2015.

[138] M. Heusel, H. Ramsauer, T. Unterthiner, B. Nessler and S. Hochreiter, "GANs Trained by a Two Time-Scale Update Rule Converge to a Local Nash Equilibrium," *Conference: Neural Information Processing Systems,* p. 6629–6640, 2017.

[139] R. Prajit, Z. Barret and V. Quoc, "Searching for Activation Functions," *arXiv,* 2017.

[140] I. Goodfellow, Y. Bengio and A. Courville, "Machine Learning Basics," in *Deep Learning*, Boston, MIT Press, 2016, pp. 108-118.

[141] G. Huang, Z. Liu, L. van der Maaten and K. Weinberger, "Densely Connected Convolutional Networks," *arXiv,* 2016.

[142] F. Such, S. Sah, M. Dominguez, S. Pillai, C. Zhang, A. Michael, N. Cahill and R. Ptucha, "Robust Spatial Filtering With Graph Convolutional Neural Networks," *IEEE Journal of Selected Topics in Signal Processing,* vol. 11, no. 6, p. 884–896, 2017.

[143] L. Zhang, J. Li, i. Zhang, H. Han, B. Liu, J. Yang and Q. Wang, "Automatic cataract detection and grading using Deep Convolutional Neural Network," *2017 IEEE 14th International Conference 5/16/2017 - 5/18/2017,* p. 60–65, 2017.

[144] M. Siar and M. Teshnehlab, "Brain Tumor Detection Using Deep Neural Network and Machine Learning Algorithm," *9th International Conference 2019,* p. 363–368, 2019.

[145] S. Kant and M. Srivastava, "Towards Automated Tuberculosis detection using Deep Learning," *IEEE Symposium Series 11/18/2018 - 11/21/2018,* p. 1250–1253, 2018.

[146] Z. Hu, J. Tang, Z. Wang, K. Zhang, L. Zhang and Q. Sun, "Deep learning for image-based cancer detection and diagnosis − A survey," *Pattern Recognition,* vol. 83, p. 134–149, 2018.

[147] S. Miao, J. Wang and R. Liao, "A CNN Regression Approach for Real-Time 2D/3D Registration," *IEEE TRANSACTIONS ON MEDICAL IMAGING,* vol. 35, no. 5, p. 1352–1363, 2016.

[148] S. Miao, J. Wang and R. Liao, "Convolutional Neural Networks for Robust and Real-Time 2-D/3-D Registration," in *Deep learning for medical image*, London, UK, Academic Press, 2017, p. 271–296.

[149] H. Pinheiro, T. Ren, S. Scheib, A. Rosselet and S. Thieme-Marti, "2D/3D Megavoltage Image Registration Using Convolutional Neural Networks," *arXiv.org,* vol. 1811.11816v1, 2018.

[150] M. Foote, B. Zimmerman, A. Sawant and S. Joshi, "Real-Time 2D-3D Deformable Registration with Deep Learning and Application to Lung Radiotherapy Targeting," *Lecture Notes in Computer Science,* vol. 11492, pp. 265-276, 2019.

[151] H. Liao, W. Lin, J. Zhang, J. Zhang, J. Luo and S. Zhou, "Multiview 2D/3D Rigid Registration via a Point-Of-Interest Network for Tracking and Triangulation," *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR),* p. 12630–12639, 2019.

[152] R. Szeliski, "Stereo correspondence," in *Computer Vision Algorithms and Applications*, London, Springer, 2011, pp. 471-500.

[153] NDI, "Passive Polaris Vega User Guide," 2016.

[154] P. Lax and M. Terrell, Calculus With Applications, Second edition ed., New York: Springer, 2014, pp. 212-213.

[155] N. Crawford, N. Johnson and N. Theodore, "Ensuring navigation integrity using robotics in spine surgery," *Journal of robotic surgery,* vol. 14, no. 1, p. 177–183, 2020.

[156] A. Menezes and T. Lindley, "Craniovertebral Abnormalities and Their Neurosurgical Management," in *Schmidek and Sweet Operative Neurosurgical Techniques*, Elsevier, 2012, pp. 2055-2070.

[157] A. Jain, R. Kon, Y. Zhou and G. Fichtinger, "C-arm Calibration - Is it Really Necessary?," *Medical image computing and computer-assisted intervention : MICCAI,* vol. 8, no. Pt 1, pp. 639-646, 2005.

[158] G. Chintalapani, A. Jain and R. Taylor, "Statistical Characterization of C-arm Distortion with Application to Intra-Operative Distortion Correction," *Medical Imaging 2007: Visualization and Image-Guided Procedures,* p. 65092Y, 2007.

[159] G. Wolberg, Digital Image Warping, Washington DC: IEEE Computer Society Press, 1994.

[160] R. Hofstetter, M. Slomczykowski, M. Sati and L. Nolte, "Fluoroscopy as an Imaging Means for Computer-Assisted Surgical Navigation," *Computer Aided Surgery,* vol. 4, no. 2, pp. 65-76, 1999.

[161] S. Schumann, X. Dong, M. Puls, L. Nolte and G. Zheng, "Calibration Of C-Arm for Orthopedic Interventions Via Statistical Model-Based Distortion Correction and Robust Phantom Detection," *IEEE International Symposium on Biomedical Imaging,* pp. 1204-1207, 2012.

[162] H. Livyatan, Z. Yaniv and L. Joskowicz, "Robust Automatic C-Arm Calibration for Fluoroscopy-Based Navigation: A Practical Approach," *Medical Image Computing and Computer-Assisted Intervention — MICCAI,* pp. 60-68, 2002.

[163] A. Telea, "An Image Inpainting Technique Based on the Fast Marching Method," *Journal of Graphics Tools,* vol. 9, no. 1, pp. 23-34, 2004.

[164] L. Desbat, R. Clackdoyle, L. Grezes-Besset, C. Mennessier and I. Bricault, "Cone-Beam Imaging of Delta Functions," *IEEE Nuclear Science Symposium,* p. 2859–2863, 2006.

[165] P. Roache, "Perspective: A Method for Uniform Reporting of Grid Refinement Studies," *Journal of Fluids Engineering,* vol. 116, no. 3, pp. 405-413, 1994.

[166] C. Mennessier and R. Clackdoyle, "Automated geometric calibration and reconstruction in circular cone-beam tomography," *IEEE Nuclear Science Symposium,* p. 5081–5085, 2008.

[167] R. Warne, "Probability and the Central Limit Theorem," in *Statistics for the Social Sciences*, Cambridge, Cambridge University Press, 2018, p. 127–150.

[168] S. Miao, R. Liao, J. Lucas and C. Chefd'hotel, "Toward Accurate and Robust 2-D/3-D Registration of Implant Models to Single-Plane Fluoroscopy," in *Augmented Reality Environments for Medical Imaging and Computer-Assisted Interventions*, Berlin, Heidelberg, Springer Berlin Heidelberg, 2013, pp. 97-106.

[169] J. Alvarez-Gomez, H. Roth and J. Wahrburg, "Comparison of Similarity Measurements and Optimizers for Intraoperative Registration of 2D C-arm Images with Preoperative CT Data in Computer-Assisted Spine Surgery," in *CURAC 2019 (18th Annual Meeting of the German Society for Computer- and Robot-Assisted Surgery)*, Reutlingen, 2019.

[170] D. Sabaei, J. Erkoyuncu and R. Roy, "A Review of Multi-criteria Decision Making Methods for Enhanced Maintenance Delivery," *Procedia CIRP,* vol. 37, pp. 30-35, 2015.

[171] C. Gendrin, H. Furtado, C. Weber, C. Bloch, M. Figl, S. Pawiro, H. Bergmann, M. Stock, G. Fichtinger, D. Georg and W. Birkfellner, "Monitoring tumor motion by real time 2D/3D registration during radiotherapy," *Radiotherapy and oncology : journal of the European Society for Therapeutic Radiology and Oncology,* vol. 102, no. 2, p. 274–280, 2012.

[172] J. Schmid and C. Chênes, "Segmentation of X-ray Images by 3D-2D Registration Based on Multibody Physics," in *Computer Vision - ACCV 2014*, Springer, 2015, pp. 674-687.

[173] ITK, "Create DRR image from Chest CT scan," ITK, Octuber 2018. [Online]. Available: https://discourse.itk.org/t/create-drr-image-from-chest-ct-scan/1370. [Accessed 26. 12. 2020].

[174] MeVisLab, "DRR," MeVisLab, [Online]. Available: https://mevislabdownloads.mevis.de/docs/current/MeVisLab/Standard/Documentation/Publish/ModuleReference/DRR.html. [Accessed 26. 12. 2020].

[175] G. Jimenez, "Development of a Digitally Reconstructed Radiograph (DRR) Generator Module using parallel computing under the MeVisLab software framework," Master's Thesis, University of Siegen,, Siegen, 2020.

[176] S. Cook, CUDA Programming: A Developer's Guide to Parallel Computing with GPUs, San Francisco: Morgan Kaufmann Publishers Inc., 2012.

[177] B. Caulfield, "What’s the Difference Between a CPU and a GPU?," 16. 12. 2009. [Online]. Available: https://blogs.nvidia.com/blog/2009/12/16/whats-the-difference-between-a-cpu-and-a-gpu/. [Accessed 26. 12. 2009].

[178] NVIDIA, "CUDA C ++ PROGRAMMING GUIDE," NVIDIA, 2020.

[179] B. Tins, "Technical aspects of CT imaging of the spine," *Insights into imaging,* p. 349–359, 2010.

[180] P. Westfall, "Kurtosis as Peakedness, 1905 - 2014. R.I.P," *The American statistician,* vol. 68, no. 3, p. 191–195, 2014.

[181] J. Pallant, A step by step guide to data analysis using SPSS for Windows, Maidenhead: Open University Press, 2007.

[182] H. Kim, "Statistical notes for clinical researchers: assessing normal distribution (2) using skewness and kurtosis," *Restorative dentistry & endodontics,* vol. 38, no. 1, pp. 52-54, 2013.

[183] D. Wheeler, Advanced topics in statistical process control: The power of Shewhart's charts, 2nd ed. ed., Knoxville, Tenn.: SPC Press, 2004.

[184] J. Alvarez-Gomez, A. Spieß, H. Roth and J. Wahrburg, "A Deep-Learning Approach to Detect Fiducials in Planar X-Ray Images for Undistortion of Conventional C-Arm Images," *Current Directions in Biomedical Engineering,* vol. 6, no. 3, pp. 40-43, 2020.

[185] L. Shapiro and G. Stockman, Computer vision, Upper Saddle River, N.J.: Prentice Hall, 2001.

[186] J. Canny, "A Computational Approach to Edge Detection," *IEEE Transactions on Pattern Analysis and Machine Intelligence,* Vols. PAMI-8, no. 6, p. 679–698, 1986.

[187] Y. Lecun, L. Bottou, Y. Bengio and P. Haffner, "Gradient-based learning applied to document recognition," *Proceedings of the IEEE,* vol. 86, no. 11, p. 2278–2324, 1998.

[188] A. Toshev and C. Szegedy, "DeepPose: Human Pose Estimation via Deep Neural Networks," *IEEE Conference on Computer 6/23/2014 - 6/28/2014,* p. 1653–1660, 2014.

[189] J. Gu, Z. Wang, J. Kuen, L. Ma, A. Shahroudy, B. Shuai, T. Liu, X. Wang, L. Wang, G. Wang, J. Cai and T. Chen, "Recent Advances in Convolutional Neural Networks," *arXiv.org,* 2015.

[190] J. Alvarez-Gomez, H. Roth and J. Wahrburg, "Intraoperative Registration of 2D C-arm Images with Preoperative CT Data in Computer Assisted Spine Surgery: Motivation to Use Convolutional Neural Networks for Initial Pose Generator," *Biomedical Engineering / Biomedizinische Technik,* vol. 64, no. s2, pp. 12-15, 2019.

[191] A. Varnavas, T. Carrell and G. Penney, "Fully automated 2D-3D registration and verification," *Medical Image Analysis,* vol. 26, no. 1, pp. 108-119, 2015.

[192] H. Zunair, A. Rahman, N. Mohammed and J. Cohen, "Uniformizing Techniques to Process CT scans with 3D CNNs for Tuberculosis Prediction," arXiv, 2020.

[193] I. Kandel and M. Castelli, "How Deeply to Fine-Tune a Convolutional Neural Network: A Case Study Using a Histopathology Dataset," *Applied Sciences,* vol. 10, no. 10, p. 3359, 2020.

[194] J. Voght, Quaternion Algebras, Springer International Publishing, 2021, p. 833.

[195] S. Grossman, Elementary linear algebra, Belmont, California: Wadsworth Publishing Company, 1987.

[196] G. Strang, Linear Algebra and Its Applications (4ed), Brooks Cole, 2005.

[197] D. Margalit and J. Rabinoff, Interactive Linear Algebra, Georgia: Georgia Institute of Technology, 2017.

[198] D. Montgomery, E. Peck and G. Vining, Introduction to Linear Regression Analysis, New Jersey: Wiley, 2012.

[199] 3D Systems, Inc, "Stereolithography Interface Specification," 1989.