



UNIVERSITY OF TWENTE.

Orthopedisch Centrum Oost Nederland
ZGT Hengelo

The Simultaneous Diagnosis of Malalignment and Liner Wear in Problematic Primary Total Knee Prosthesis with Low-Field MRI

L. van den Hoven
MSc. Thesis
December 2018



ocon
Orthopedische kliniek

Supervisors:

Prof. dr. ir. N.J.J. (Nico) Verdonchot
Dr. R. (Rianne) Huis in 't Veld
Dr. A.J.H. (Anne) Vochteloo
Dr. ir. F.F.J. (Frank) Simonis
B.J.C.C. (Bregje) Hessink-Sweep, MSc.

1 Abstract

Total knee arthroplasty is a cost-effective and frequently performed procedure for the treatment of late-stage osteoarthritis. However, up to 8.5% of primary total knee prosthesis implants require revision surgery. It is important to understand the underlying pathology to reduce the chance of recurring problematic knee prosthesis. To determine the presence or absence of possible underlying pathologies of problematic total knee prosthesis, many different imaging modalities are required. It is hypothesized that low-field MRI, due to its inherent lower susceptibility artifacts, can provide an alternative to simultaneously diagnose the pathologies of the problematic primary total knee prosthesis. In this study, a method to determine the malalignment and liner wear have been developed. The results of the measurements show some correspondence with the reference values proposed in the literature. Also, a difference between standing and supine imaging of the patient is indicated in this study. However, to provide clinically relevant data, further research is required. Future research should focus on increasing image quality, increasing study population, especially directly post-operative patients for direct comparison with the literature, and the indicated differences between standing and supine imaging should be investigated further.

2 List of acronyms

TKA	Total Knee Arthroplasty
UKA	Unicompartmental Knee Arthroplasty
TKP	Total Knee Prosthesis
PE	Polyethylene
cFA	Coronal Femoral Angle
cTA	Coronal Tibial Angle
sFA	Sagittal Femoral Angle
sTA	Sagittal Tibial Angle
CBA	Component Balance Angle
TFA	Tibiofemoral Angle
US	Ultrasound
CT	Computed Tomography
RSA	Radio Stereophotogrammetric Analysis
MRI	Magnetic Resonance Imaging
MR	Magnetic Resonance
RF	Radiofrequency
FID	Free Induction Decay
OCON	Orthopedisch Centrum Oost Nederland (Orthopaedic Centre East Netherlands)
GCS	Global Coordinate System
LCS	Local Coordinate System
SVD	Singular Value Decomposition
CI	Confidence Interval
sd	Standard Deviation

Contents

1	Abstract	2
2	List of acronyms	3
3	Introduction	5
3.1	The Pathological Knee	5
3.1.1	Osteoarthritis	5
3.1.2	Total Knee Arthroplasty	5
3.2	Pathological TKP	7
3.2.1	Malalignment	7
3.2.2	Polyethylene liner wear	9
3.3	Goal of the Study	9
3.4	Study Design	10
4	Theory	12
4.1	Common Orthopaedic Imaging	12
4.2	MRI	12
4.2.1	Susceptibility Artifacts	13
4.2.2	T2 Relaxation	14
4.3	Low-field MRI	15
4.4	Overview	16
5	Data Acquisition and General Analysis	17
5.1	Materials and Methods	17
5.1.1	MRI Data Acquisition	17
5.1.2	3D Scans	18
5.1.3	Segmentation	19
5.1.4	Registration	19
5.2	Results	20
5.2.1	Patients	20
5.2.2	MRI Data Artifacts	20
5.2.3	Segmentation	21
5.2.4	Registration	24
5.3	Discussion	25
5.3.1	MRI Data Acquisition	25
5.3.2	3D Scans	25
5.3.3	Segmentation	26
5.3.4	Registration	26
6	Specific Analysis: Malalignment	27
6.1	Materials and Methods	27
6.2	Results	33
6.3	Discussion	37
7	Specific Analysis: Liner Wear	38
7.1	Materials and Methods	38
7.2	Results	42
7.3	Discussion	45
8	General Discussion	47
9	Recommendations	49
10	Conclusion	50
11	Appendix A: Liner Wear Python Code	57
12	Appendix B: Malalignment Python Code	71

3 Introduction

3.1 The Pathological Knee

The knee joint is comprised of the two major bones that constitute the leg, the femur (upper leg bone) and the tibia (lower leg bone) and is accompanied by the patella. As such it is an instrumental part of the walking motion [1]. Figure 1 shows the normal anatomy of the knee, including the muscles and tendons that are involved in maintaining knee mobility and stability. Pathology of the knee undermines the integrity of the normal knee causing dysfunction. One of the most common pathologies of the knee, and the most common joint disorder in the world, is osteoarthritis [2, 3]. Many factors influence the development of osteoarthritis, such as obesity, joint injury, bone alignment and physical activity [2–5]. Furthermore, a strong correlation between increasing age and the prevalence of osteoarthritis has readily been shown, making age a significant benefactor for the development of osteoarthritis [4].

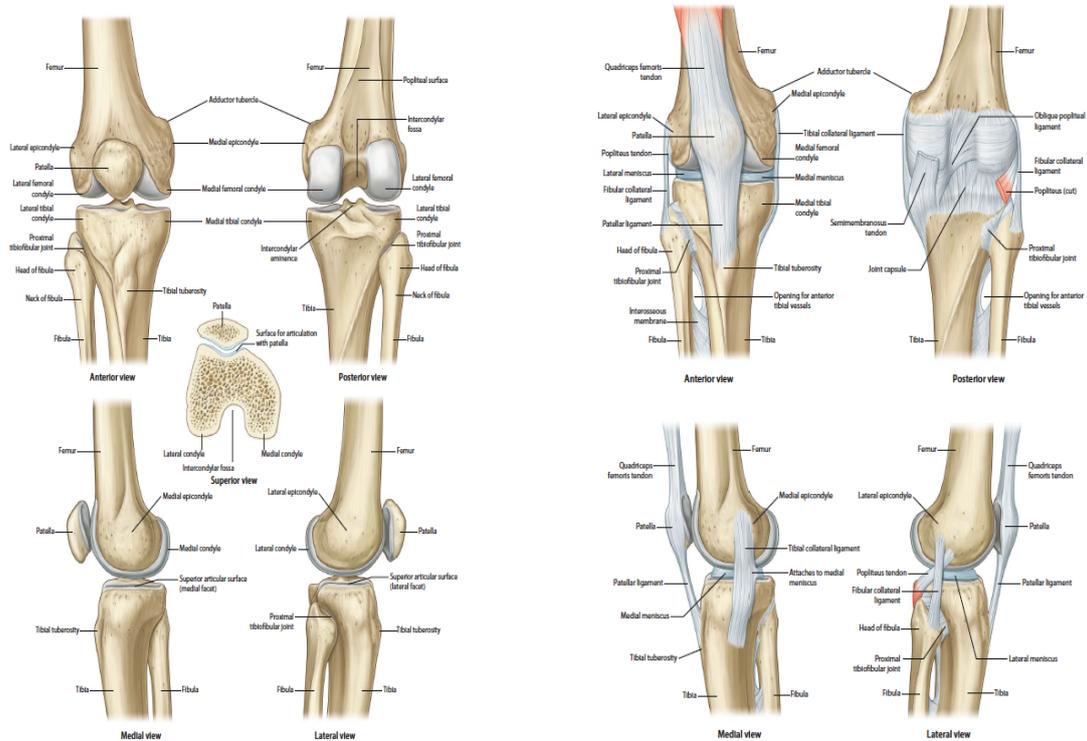


Figure 1: The bony and cartilaginous structures of the knee. (From Gray’s Anatomy [6]).

3.1.1 Osteoarthritis

Osteoarthritis has a major impact on the function of the knee joint. While the normal knee has a full range of motion of $0 - 160^\circ$ [7, 8], with the normal range between $0 - 135^\circ$ [9], it has been shown that osteoarthritis significantly restricts this motion, resulting from the pathological changes in the knee joint [10]. The main symptoms of osteoarthritis are joint pain and stiffness caused by a diverse array of pathologies including focal damage and loss of articular cartilage, osteophytes, ligamentous laxity, periarticular muscle atrophy, and synovial distension and inflammation [11]. The joint pain and stiffness result in decreased knee joint function and overall activity. The treatment for osteoarthritis is not curative and consists initially of non-surgical methods such as physiotherapy and pain control. Surgical intervention is considered only for later stages of osteoarthritis or when the non-surgical options have been exhausted [3].

3.1.2 Total Knee Arthroplasty

For late-stage osteoarthritis, arthroplasty is the surgical option. Total knee arthroplasty (TKA), and unicompartmental knee arthroplasty (UKA) for less extensive osteoarthritis is a very often

performed surgical procedure for the treatment of the osteoarthritic knee, during which the pathological biological knee is replaced with a total knee prosthesis (TKP). In 2016, a total of 27.918 primary total knee arthroplasties were performed in the Netherlands, of which 96% for osteoarthritis [12]. With an ever aging population, the yearly incidence of osteoarthritis will continue to rise and it is expected that the current prevalence of TKA procedures will increase by 600% in 2030 [13,14].



Figure 2: Illustration of how a TKP will look when all components are combined (The patella button is not shown here). From [15].

The current TKA procedure is an effective, safe, cost-effective and durable treatment method for the treatment of osteoarthritis [16–22]. Depending on the extent of the osteoarthritis, the TKP consists of three or four components, the femoral component, tibial component, polyethylene (PE) liner, and, if necessary, a patella button (see also figure 3). During surgery, the pathological bone fragments are resected and the femoral, tibial and patellar bone surfaces are prepared to fit the prosthesis components [23]. To provide correct placement and alignment, the required cuts for a correct prosthesis component fit are made using special tools for knee arthroplasty surgery [23].

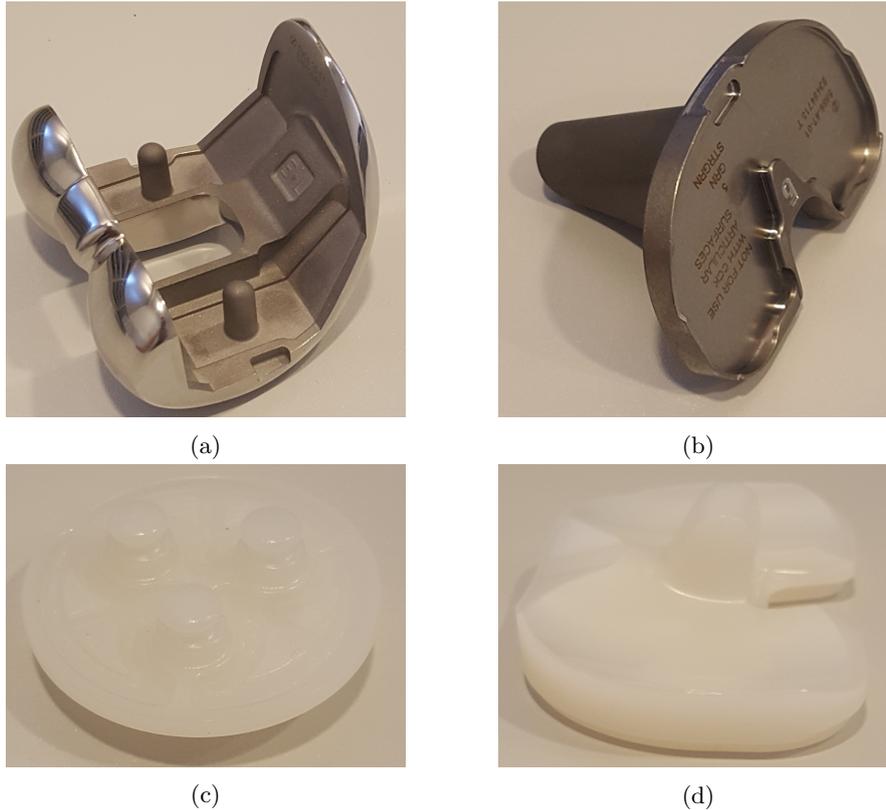


Figure 3: Examples of the individual components that comprise a TKP; The (a) femoral component, (b) tibial component, (c) patella button, and (d) PE liner.

3.2 Pathological TKP

Despite the efficacy of the TKA procedure for the treatment of osteoarthritis, up to 8.5% of implants require revision, more than 25% of which within 5 years [20]. TKP revision surgery is a very complex, demanding and expensive surgery which is ideally avoided. To avoid failure of the revision prosthesis due to similar pathology, the planning of the revision surgery should consider the failure mechanism of the primary TKP [24–26]. Therefore, it is important to correctly diagnose the mechanism of failure of the TKP.

Various pathologies preceding a revision of one or more components of the knee prosthesis have been identified [27]. The main reasons for total knee prosthesis (TKP) revision include: aseptic loosening (23.2-39.9%), infection (16.2-27.4%), instability (7.5-25.1%), patellofemoral complications (21.1%), polyethylene liner (PE) wear (7.4-18.1%), malalignment(2.9-14.0%), arthrofibrosis (4.3-6.9%) and periprosthetic fractures (1.4-4.7%) [12,28–30]. For this study, the focus is on the diagnosis of polyethylene liner wear and malalignment. These failure mechanisms require surgical prosthesis component or liner revision, which has already been described as being a difficult and time-consuming procedure. Furthermore, these processes are both currently evaluated in the clinic with standard x-ray imaging, which means these pathologies may benefit the most from a new imaging standard [31,32]. In the following section, the pathology of malalignment and liner wear will be described to provide further understanding of the processes.

Pathology Background

3.2.1 Malalignment

Malalignment is one of the main causes of aseptic loosening and subsequent revision surgery and often induces wear or osteolysis. Malalignment is described as a varus-valgus angle of the mechanical axis of the knee of more than 3°varus or 7°valgus [32–34]. Many studies have debated whether mechanical, anatomical or kinematic alignment of the knee is the best determinant for positive outcomes, and currently, there exists no single guideline for correct TKA alignment [32, 34, 35].

The most common convention during surgery is to align the prosthesis components within 3° of the anatomical axis, as it has been described that an angle beyond 3° of varus or 7° of valgus is associated with a more rapid failure and less adequate results after TKA [35,36]. A correct alignment, therefore, is an important parameter and should be considered when determining the origin of residual pain after TKA.

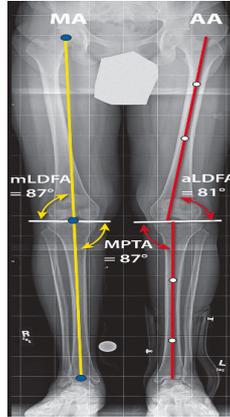


Figure 4: Standing x-ray showing the mechanical and anatomical axes of the femur and tibia as well as the standard values for the angles. The difference between the mechanical femoral axis (MAF) and anatomical femoral axis (AAF) is typically between 5° and 7° (6° in this image). [32] MA: Mechanical axis, AA: anatomical axis, mLDFA: mechanical lateral distal femoral angle, aLDFA: anatomical lateral distal femoral angle, MPTA: medial proximal tibial angle.

Measuring the alignment of a TKP is normally performed based on roentgenographic inspection. The standard methods consist of a set of angle measurements between the anatomical axis of the tibia or femur and the corresponding axis of the tibial or femoral prosthesis component as shown in figure 5.

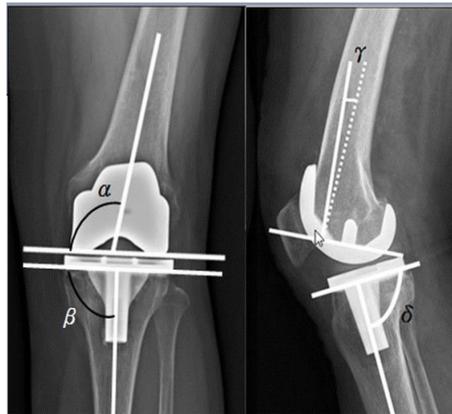


Figure 5: Angles that are determined with radiography for the assessment of malalignment of the femoral and tibial components of a total knee prosthesis according to the total knee arthroplasty roentgenographic evaluation and scoring system [37]. From left to right, the measured angles are: coronal Femoral Angle (cFA, α), coronal Tibial Angle (cTA, β), sagittal Femoral Angle (sFA, γ), and sagittal Tibial Angle (sTA, δ). (Image source: Lee *et al.* (2014) [38]. The tibio-femoral angle (TFA) is not shown here.

From the total knee arthroplasty roentgenographic evaluation and scoring system [37], several angles are defined that are important for the evaluation of malalignment of a TKP. The coronal femoral angle (cFA), coronal tibial angle (cTA), sagittal femoral angle (sFA), sagittal tibial angle (sTA), and the tibio-femoral angle (TFA). The coronal femoral angle (cFA) is defined as the angle between the anatomical axis of the femur and the transverse-plane of the femoral prosthesis (angle α in figure 5). Similarly, the coronal tibial angle (cTA) is defined as the angle between the

anatomical axis of the tibia and the transverse-plane of the tibial prosthesis (angle β in figure 5. The angle between the anatomical axis of the femur and the transverse-plane of the femoral prosthesis component determines the sagittal femoral angle (sFA), and the same approach determines the sagittal tibial angle (sTA)(angle γ and δ in figure 5, respectively). The tibio-femoral angle (TFA) is determined by the sagittal angle between the anatomical axes of the tibia and the femur. Lastly, the balance of the tibial and femoral prosthesis components has been shown to impact the stability and wear pathology [39].

3.2.2 Polyethylene liner wear

Polyethylene (PE) liner wear constitutes the wear of the liner of a prosthesis due to shear stresses resulting in periprosthetic osteolysis. This inflammatory process occurs due to excessive PE particles that activate a cascade of biological processes, resulting in bone resorption and osteolysis [40–42]. Liner wear mechanisms such as pitting and burnishing (fig 6) illustrate the damage that can be done to prosthesis liners if left untreated [43, 44].

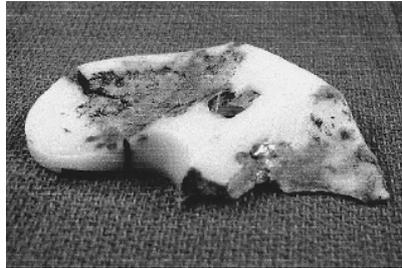


Figure 6: Result of liner wear mechanisms such as pitting and burnishing. Destruction of the liner due to wear results in loose particles, resulting in osteolytic processes [43].

Considering the significant contribution of liner wear to late-stage TKA failure [45] imaging of the extent of the liner wear is very important. Currently, standing anteroposterior radiographic images are used to determine liner wear, however, practical limitations such as positioning of the knee decrease the clinical relevance of these assessments [32]. Furthermore, while severe liner wear can be indicated by an x-ray, mild to moderate liner wear tends to be difficult to evaluate [31]. As such, various methods have been proposed to more accurately assess the state of the liner *in vivo*, such as radiostereophotogrammetric analysis based radiographic assessment [31], sonography or ultrasound (US) [46, 47] and more recently, MRI [48]. These methods all determine the thickness of the liner compared to some previously determined parameter, *e.g.* the knowledge of the liner thickness (from a digital model), an earlier radiographic image, or evaluation of the liner by an expert. In this study the liner wear will be determined by calculating the distance between the tibial and femoral prosthesis components *in vivo*.

3.3 Goal of the Study

The previous sections show that the pathological mechanisms for the failure of a TKP are of importance for further treatment. The treatment path for *e.g.* aseptic loosening, is different from that of patellofemoral pain or PE wear [49–51]. However, rejection or confirmation of a differential diagnosis requires varying imaging examinations, because, as can be seen in figure 7, no diagnostic imaging method currently used for the analysis of a knee prosthesis, has the capabilities to simultaneously determine the presence or absence of all of the above pathologies.

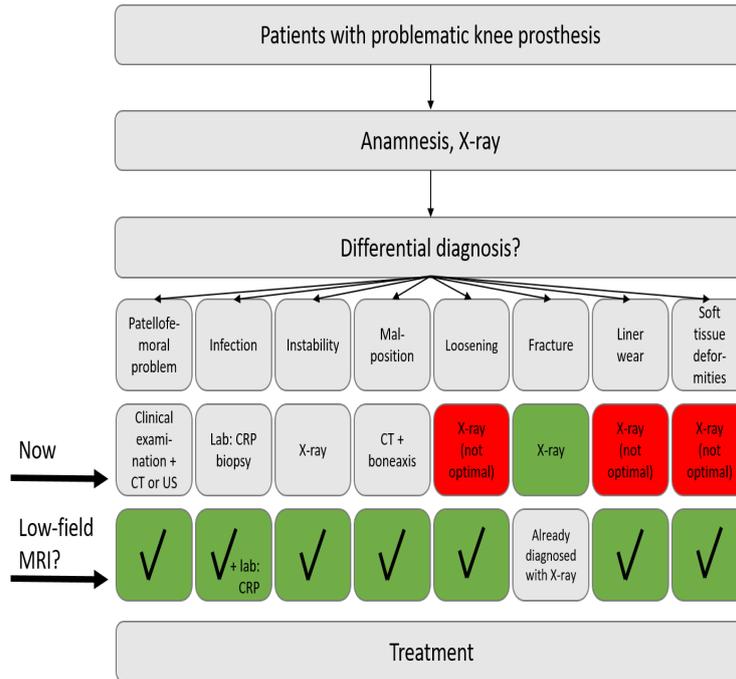


Figure 7: Overview of the methods available for the diagnosis of problematic total knee prosthesis pathologies (from MRIPro voucher request).

Magnetic resonance imaging (MRI) has the ability to image all tissues of the body and low-field MRI is expected to reduce the metal artifacts in MRI data. Therefore, low-field MRI may provide an alternative imaging method for the simultaneous diagnosis of problematic TKA pathologies, and liner wear and malalignment in particular.

The main research question for this study is: Can low-field MRI provide an alternative for the current clinical practice, for the simultaneous diagnosis of malalignment and liner wear for problematic total knee prosthesis patients?

This research question will be divided into the following sub-questions:

Can low-field MRI determine prosthesis component malalignment of the problematic total knee prosthesis?

Can low-field MRI determine the polyethylene liner wear of the problematic total knee prosthesis?

3.4 Study Design

This report describes the methods developed to provide answers to the above research questions. The clinical relevance of this study has been described in the above sections. To provide an understanding of the materials used and decisions made in this study, the **Theory** section gives an introduction to low-field MRI and the problems with imaging metallic artifacts as well as an overview of other imaging methods used in orthopaedics. After a theoretical basis has been provided, the methods for this study will be described in a threefold manner. Figure 8 shows the overall structure of the study, consisting of three parts, **Data Acquisition**, **General Analysis**, and **Specific Analysis**. The specific analysis in turn is split in malalignment and liner wear measurements.

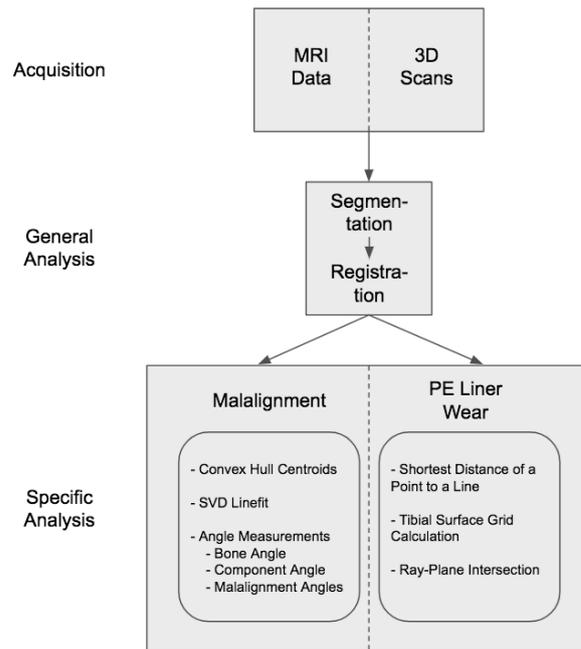


Figure 8: Flowchart of the study.

After the **Theory** section, the data acquisition and General Analysis parts of the study will be presented. The materials and methods, results and discussion for the data acquisition and general analysis will be discussed here. Subsequently, the specific analysis will be split in malalignment and liner wear measurements, each with their own method, results and discussion sections. This structure was chosen to improve readability of the report.

4 Theory

In this section, the imaging modalities used for the diagnosis of various pathologies of the problematic total knee prosthesis are presented.

4.1 Common Orthopaedic Imaging

Various imaging modalities are used for the diagnosis of problematic total knee arthroplasty (TKA) pathologies. X-ray, ultrasound(US), computed tomography(CT), röntgen stereophotogrammetric analysis (RSA), and magnetic resonance imaging (MRI) can be employed to determine different underlying pathologies of TKA problems. Each imaging method provides advantages and disadvantages, described here and shown in the overview (table 1) at the end of this section.

X-ray imaging of the knee is routinely used for the assessment of the components of the total knee prosthesis. Some strong advantages for x-ray imaging is that it does not suffer from pronounced metal artifacts, is fast, and is widely used in orthopaedics, and clinical and research settings. The major disadvantage of X-ray imaging is that it is a 2D imaging method and it is difficult to correctly position the region of interest to obtain an ideal image for objective measurements. Also, performing an X-ray exposes the patient to radiation, which is ideally avoided. [52]

Ultrasound (US) imaging in orthopaedics can be used for the assessment of the prosthesis liner. However, the penetration depth is very low, due to scatter and absorption of the ultrasound waves by various tissues, especially dense tissues. Therefore, US is not very useful for the imaging of structures deep in the knee.

Computed tomography (CT) is a fast and accessible imaging technique for the imaging of the musculoskeletal system, among others. CT is also a Röntgen based technique, however, in contrast to the x-ray, CT is a 3D imaging method. A major disadvantage of CT, and Röntgen imaging methods in general is that it has low soft tissue contrast. Furthermore, metal artifacts are present *e.g.* in the form of streak artifacts near and between metallic objects in the imaged volume. [53]

Röntgen stereophotogrammetric analysis (RSA), is an analytical imaging method specifically used for the measurement of prosthetic migration. This method is very accurate and is often used for the measurement of micro-motion to predict and monitor aseptic prosthesis loosening. The major disadvantage of this method is the requirement of per-operatively placing peri-prosthetic tantalum markers, which makes this analysis method available in a research setting only. [54–58].

4.2 MRI

Magnetic resonance imaging (MRI) is an imaging technique widely used in clinical practice. It can image all of the various tissues in the body, making it very suitable for the imaging and diagnosis of musculoskeletal pathologies [59]. The following section provides background information about the MRI technique and an introduction to low-field MRI.

Many molecules in the body have magnetic properties, however, for standard MR imaging, the magnetic properties of bound hydrogen (^1H) nuclei are used to obtain images. Each hydrogen nucleus, or proton, in the body possesses a property called spin, or spin angular momentum. In a normal setting, all the spins are oriented randomly in all directions, setting the average of all the spins, referred to as the net magnetization (\mathbf{M}), to 0 [60]. When an external magnetic field (assumed to be homogeneous) is applied, such as in an MRI scanner, the spins align along the direction of the magnetic field lines and the net magnetization consequently becomes larger.

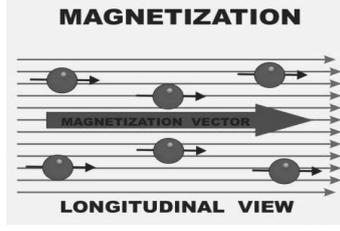


Figure 9: The figure above shows the direction of the magnetization vector, parallel to the main magnetic field lines from the \mathbf{B}_0 field. [61]

While \mathbf{M} is a three-dimensional vector consisting of x-, y-, and z-directional components, when a homogeneous external magnetic field is present, the x-, and y-components are 0 and only its z-component is present and at its maximum, $\mathbf{M}_z = M_0$. After excitation by a radio-frequency (RF) pulse the x-, and y-components of \mathbf{M} become larger. During an MRI scan, \mathbf{M} is excited by an RF pulse, causing it to rotate away from its original orientation, \mathbf{M}_z , by an angle α , known as the flip angle, to \mathbf{M}_{xy} . The initially coherent spins of \mathbf{M} will then start to rotate or precess around \mathbf{B}_0 by angle α and with frequency:

$$f_{\text{larmor}} = \gamma B_0 \quad (1)$$

where γ is the gyromagnetic ratio of a proton, which is equal to $\gamma = 42.58 \text{ MHz/T}$.

4.2.1 Susceptibility Artifacts

For many MRI applications, it is crucial to supply spatial information of the volume. To provide this spatial encoding, gradients are used. When applying a gradient to the \mathbf{B}_0 field, the field strength increases from $\mathbf{B}_0 - z\mathbf{G}_z$ to $\mathbf{B}_0 + z\mathbf{G}_z$, with \mathbf{G}_z the change in field strength per unit of length (*e.g.* 1mT/cm), z the distance from the center of the gradient. In MRI, an individual slice can be excited by an RF-pulse that corresponds to the specific resonance frequency of the protons in that slice:

$$f = \gamma(B_0 \pm zG_z) \quad (2)$$

However, the presence of metallic objects can cause problems with this method, due to the intrinsic properties of metallic objects and MRI itself.

When metals are placed in a magnetic field, in this case the homogeneous \mathbf{B}_0 field, they induce an internal magnetic field that disturbs the \mathbf{B}_0 field lines. This causes a change in the strength of the magnetic field around the metal and consequently alters the resonance frequency of the spins in that area. The corresponding resonance frequency of the affected spins becomes

$$f = \gamma(B_0 + B_{\text{local}}) \quad (3)$$

which differs from the original larmor frequency from equation 1. The spins around a metallic object are subject to both the field gradient (\mathbf{B}_0 and the magnetic field distortion (B_{local}) of the object. When the frequency of the protons in *e.g.* slice 4 corresponds with the frequency of the protons in slice 1, the protons in slice 4 will also be excited when the RF-pulse is

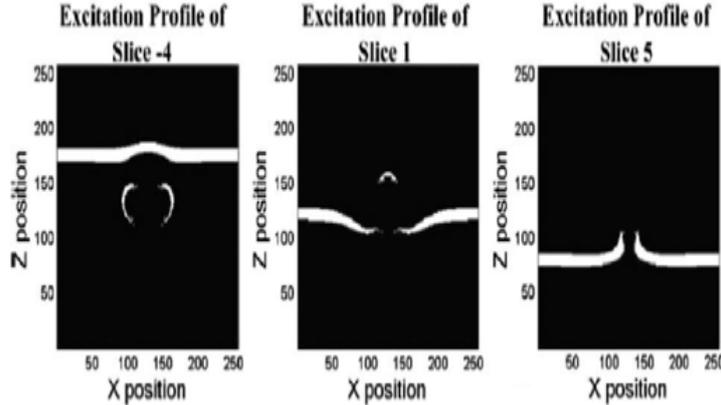


Figure 10: Effect of metal objects on the spatial encoding of a volume. The excitation profiles show the positions of the excited spins for the resonance frequency of several slices resulting from a scan with a metallic ball, viewed from above. Due to susceptibility, the white bands representing the excited spins, are not a straight line, which is expected when exciting a single slice. Furthermore, other spins not expected in the slice are also excited. [62]

The strength of the magnetic susceptibility artifact caused by the presence of a metallic object is proportional to the strength of the main magnetic field by the relation:

$$\Delta r \propto \frac{k\Delta\chi B_0}{G} \quad (4)$$

where Δr is the pixel shift in the imaging plane, k is some proportionality constant depending on the shape of the object, $\Delta\chi$ is the difference in susceptibility between the material and the surrounding tissue, B_0 is the strength of the main magnetic field, and G is the strength of the readout gradient [63].

4.2.2 T2 Relaxation

Apart from the above relation between metallic objects and a magnetic field, another mechanism causes artifacts due to the presence of metallic objects. This mechanism is embedded in the relaxation parameter T2, that determines much of the signal measured with MRI.

It has previously been stated that an RF-pulse rotates \mathbf{M} and causes it to precess around \mathbf{B}_0 with resonance frequency f . To detect a signal from this precession we must turn to Faraday's law of induction which states that a change in magnetic flux generates an equally large, but oppositely charged electromagnetic force in a nearby conductor:

$$\varepsilon = -\frac{\delta\phi_B}{\delta t} \quad (5)$$

where ε is electromagnetic force, ϕ_B is magnetic flux, and t is time. The measurable precession of \mathbf{M} around \mathbf{B}_0 , however, does not maintain its strength infinitely due to \mathbf{M} returning to its original alignment along the \mathbf{B}_0 field lines. This process is called relaxation and is determined by two parameters, T1 and T2, and results in a free induction decay (FID) signal. The relation between magnetization, T1 and T2 is given by:

$$M_{xy} = M e^{-t/T2} \quad (6)$$

$$M_z = M_0(1 - e^{-t/T1}) \quad (7)$$

where M is the magnetization in xy-plane directly after excitation of \mathbf{M} by the RF-pulse and t is the time.

T2 relaxation is caused by two mechanisms. Firstly, energy transfer from the spins to the surrounding lattice (T1 relaxation) or to other spins (T2) causes relaxation. Secondly, each spin in \mathbf{M} is affected by different fluctuating inhomogeneities in the \mathbf{B}_0 field. These inhomogeneities are

caused by molecular interactions and are random by nature due to *e.g.* Brownian motion, electron shielding, and other mechanisms [64]. The resulting magnetic field inhomogeneities are very small, yet cause the individual spins of \mathbf{M}_{xy} to lose phase coherence and therefore causes T2 relaxation.

During an MRI scan, however, signal decay is normally much faster than the expected T2 would predict. This is due to field inhomogeneities on a larger scale. The presence of an object, such as a hand, in the scanner, will disturb the main magnetic field, causing changes in local magnetic field strength \mathbf{B}_{local} . This type of T2 relaxation is called T2*-relaxation and is defined as:

$$\frac{1}{T2^*} = \frac{1}{T2} + \gamma(B_0 \pm B_{local}) \quad (8)$$

While T2 relaxation based on the molecular interactions is random and cannot be corrected, these larger local changes are static and can be corrected. This is done by introducing a 180° RF-pulse, which flips the spins and the phase differences. The flipping of the spins and phase differences causes the spins to re-phase and create an echo. This echo is the signal that would be measured if only T2 relaxation was present. Figure 11 illustrates this principle, where the red arrow represents the magnetization vector \mathbf{M} .

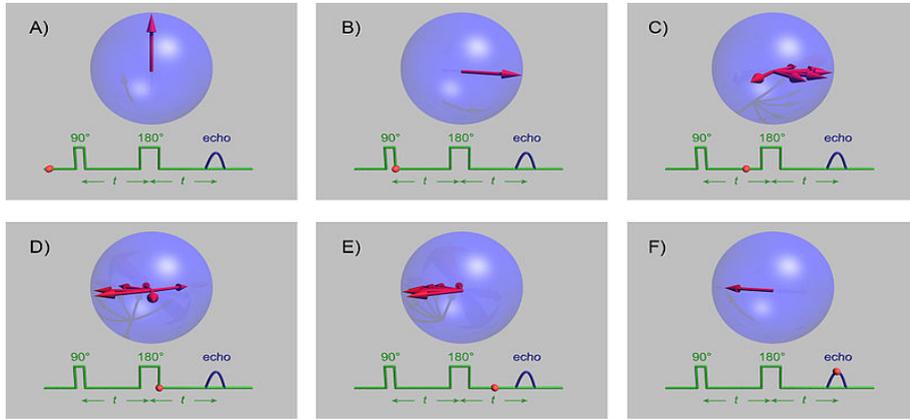


Figure 11: The illustration above shows the principle of a Spin Echo (SE). After a 90 degree RF-pulse, \mathbf{M} rotates and its spins start to de-phase. After a subsequent 180 degree RF-pulse, the spins re-phase and create an echo.(from [65])

When considering the disturbances of the \mathbf{B}_0 field by a metallic object, the difference of the susceptibility between the metallic object and the surrounding tissue must be taken into account. From Schenk *et al.* [66] it can be seen that this difference is up to 10^3 larger than the differences between water based tissues in the body. As such, the metallic objects in standard clinical 1.5T MRI have a large effect on the T2* relaxation of the signal as well as the spatial encoding of the volume. To provide lower metallic artifacts in the image data, low-field MRI was used in this study. Below, further information on low-field MRI is provided.

4.3 Low-field MRI

Standard clinical MRI scanners work with main magnetic fields in the range of 1.5 – 3.0 tesla [64]. Low-field MRI is defined as having a main magnetic field below 0.3 tesla. To see what a lower magnetic field does to the strength of the signal, the Boltzmann equation can provide some insight [60]:

$$M_0 = B_0 \frac{\gamma^2 h^2 N_s}{4kT} \quad (9)$$

When disregarding the other terms, we can instantly see that $M_0 \propto B_0$. For a 0.25T scanner, such as the one considered in this study, the expected M_0 will be at least six times smaller than that of a standard clinical MRI scanner with a magnetic field strength of 1.5 – 3.0T (with all other variables equal). This, however, does not mean that low-field MRI is six times worse than standard MRI. As has been shown by Ghazinoor *et al.* (2007) [67], the diagnostic value of low-field MRI in orthopedics is similar to standard 1.5T MRI and has a good correlation with per-operative

arthroscopic findings [68].

One resulting advantage of the lower magnetic field strength of low-field MRI is its effect on susceptibility artifacts. From the relation between the strength of susceptibility artifacts and main magnetic field strength in equation 7, we can also derive that the strength of the artifact in low-field MRI should be lower than that of standard MRI. Indeed, Farahani *et al.*(1990) [69] have also shown this is the case.

Furthermore, one of the main advantages of the low-field MRI present at the University of Twente is the availability of weight-bearing scanning. The clinical imaging for malalignment and liner wear is performed in standing position. Low-field MRI can mimic the standard radiology exams for total knee prosthesis imaging as described in literature [31, 46].

Beyond the advantage of lower strength susceptibility artifacts, low-field MRI can have several other advantages depending on specific designs:

- Open design
- Cheaper
- Lower energy deposition in tissues

4.4 Overview

In the paragraphs above, a theoretical background is provided to show how low-field MRI may affect MR image data. While MRI can image all tissues In the paragraphs above, an intuition of how MRI and specifically low-field MRI can aid the analysis of *in vivo* primary total knee prosthesis is presented. Furthermore, several other available diagnostic imaging methods have been discussed. Here, an overview of the advantages and disadvantages of the available imaging methods is given.

Table 1: A summary of the advantages and disadvantages of CT, X-ray, high-field MRI and low-field MRI for imaging TKP *in vivo*.

	Benefits	Disadvantages	Useful for
X-ray [52]	Fast, good bone imaging, no metal artifacts	2D only	Imaging bone and the metal prosthesis components
CT [53]	Fast, good bone contrast, 3D	Low soft-tissue contrast, metal artifacts	Imaging bone and the metal prosthesis components
RSA [54–58]	Very accurate	Requires per-operative placement of tantalum markers	Micro-motion/Migration
US	Fast, no radiation, mobile	Very low penetration depth	Liner wear and patellofemoral pain
High-field MRI [60]	3D, high soft-tissue contrast, can image fluids, no radiation	Metal artifacts, may provoke claustrophobia [70–72]	Imaging all tissues, including bone, soft tissues and fluids
Low-field MRI	Same as high-field MRI, intrinsic metal artifact reduction, open bore	Metal artifacts still present, low SNR [73]	Same as high-field MRI [67, 74–78]

5 Data Acquisition and General Analysis

This section will describe the materials and methods and results for the Data Acquisition and General Analysis parts of the study as well as provide a section specific discussion. The data acquisition and general analysis consist of the MRI data acquisition, the acquisition of the 3D optical scans, the segmentation and registration segments of this study.

5.1 Materials and Methods

5.1.1 MRI Data Acquisition

Primary Dataset

A primary dataset of a scanned patient with a primary total knee prosthesis was available and used as a template to determine the intra-observer variability of the image segmentation and to develop the malalignment and liner wear workflow and measurements. This dataset was not created following the protocol described below but was previously available as a sample for the purpose of development. This data is denoted as Pt. 101.

Patients

For this study, data was obtained from patients all under care at OCON Hengelo who have given informed consent for the use of their low-field MRI data to be used for this study. One patient with and six without residual pain are included. Excluded are patients with a contra-lateral TKP, an inability to stand for 15 minutes, or a contra-indication for undergoing an MRI scan. The included patients are scanned at the University of Twente at the Esaote low-field (0.25T) scanner (Esaote G-scan brio, Esaote SpA ©2017, fig12). Scans are made with knee coil 1 with the patient in a vertical position (81.0° angle). This is done to mimic the current clinical practice for radiology examination of the knee, which is a standing, load-bearing radiograph. Furthermore, patients are scanned in supine position as well. For comparison, one of the patients' supine data set is used for analysis. Patients are numbered 102-107a for the patients without complaints of the knee, 201 for the patient with complaints of the knee, and 107b for the patient data acquired in supine position.

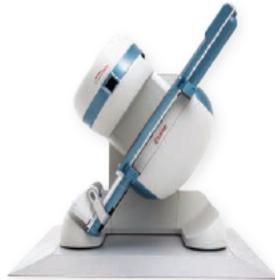
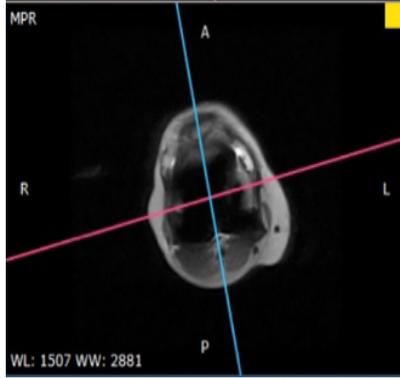


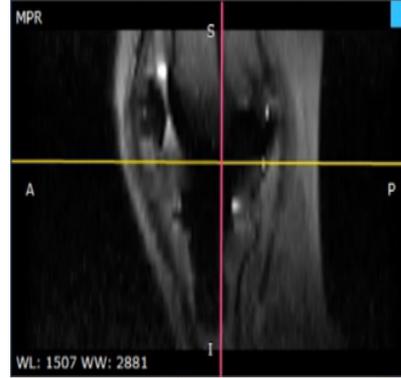
Figure 12: Low-field MRI scanner at the University of Twente (Esaote G-scan brio, Esaote SpA ©2017). Shown orientation not known, however, the figure illustrates the possibility of different angles of the scanner to allow for weight-bearing positions. (From Esaote documentation [79])

MRI Scan Protocol

Images were acquired using a T1 spin-echo sequence. Sagittal images of the data were used for further analysis with dimensions: 256×256 voxels per slice, $0.89 \times 0.89\text{mm}^2$ pixel size and a pixel spacing of 4.4mm. During the planning phase of scanning, the images are aligned, such that the coronal plane is perpendicular to the femoral condyles, as shown in figure 13a. The sagittal plane is perpendicular to the coronal plane, shown as the blue colored line in figure 13a and the pink line in figure 13b. This alignment during the planning phase of the MRI scan is performed for each patient separately, to ensure correct alignment of the relevant planes for each individual case. The coronal and sagittal planes, as described above, correspond to the global coordinate system (GCS) XZ- and YZ-planes, respectively, in both Mimics and python.



(a) Transverse plane in the planning stage of the low-field MRI scanner. The pink line represents the global x-axis and the blue line represents the global y-axis.



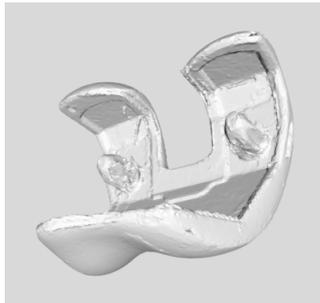
(b) Sagittal plane in the planning stage of the low-field MRI scanner. The pink line represents the global z-axis and the yellow line represents the global y-axis.

Figure 13: Alignment phase of the planning of an MRI scan.

5.1.2 3D Scans

Optical Scanning of Components

3-Dimensional models of the tibial and fibular components of the total knee prosthesis have been made with an optical 3D scanner at the University of Twente (Konica Minolta Vivid 910) and processed in Meshmixer 3.3 (©2017 Autodesk, Inc.). The total knee prosthesis components that were scanned and used for this study are the Smith and Nephew GENESIS II components. The scanned models were femur size E right and tibia size 5. Because only one set of components was available for the optical scan, the obtained femoral and tibial component scans were linearly scaled to also estimate models of the other possible sizes, using the values given in [80]. During the processing step, the prosthesis components (both femoral and tibial) are aligned with the global coordinate system as defined in the MRI data acquisition section. The resulting models are used during the registration step.



(a) Example of a femoral component of a TKP.



(b) Example of a tibial component of a TKP.

Figure 14: 3-Dimensional models of the tibia and femur components of the total knee prosthesis obtained from the 3D optical scanner.

3D Model Alignment

After acquisition of the 3D models with the optical scanner, the 3D models are aligned semi-automatically along the global z-axis. Using the integrated global coordinate system of the software, the 3D models are aligned such that the top-bottom axis of the models correspond to the global z-axis, the left-right axis of the models correspond to the global x-axis, and the front-back axis of the models corresponds to the global y-axis. Furthermore, the placement of the femoral component is above the tibial component, with a gap of X mm, where X is the thickness of the prosthetic liner.

Resulting from this alignment are femoral and tibial components in the ideal alignment. This alignment of the 3D models results in a separate set of components for each component and liner size combination. Alignment of the components was performed in Meshmixer and 3-Matic.

This alignment and positioning is performed in order for the registration to work correctly. The 3D model and the segmented data must be in each others proximity, and, ideally, aligned similarly, which is achieved in this step. If the segmented data and the 3D models are not near each other when loaded into Mimics, the registration algorithm fails.

5.1.3 Segmentation

After data acquisition, but prior to image analysis or measurements, image segmentation must be performed. It is a crucial step towards further image analysis, however a difficult task due to the metal artifacts present in the MRI data. While many (semi-)automatic segmentation algorithms exist, manual segmentation remains the gold standard for image segmentation [81, 82]. Therefore, manual image segmentation will be performed to obtain segmentation models of the relevant structures from the MRI data. Segmentation will be performed in Mimics Innovation Suite 19 (Copyright Materialise 2017). During the segmentation step, the tibia, femur, tibial prosthesis component, and femoral prosthesis component are segmented. The resulting segmentations will be rendered as 3-dimensional models to be used for later registration and analysis.

To determine the variability of manual segmentation with Mimics for segmenting the femur, tibia, femoral prosthesis component, and tibial prosthesis component, three separate segmentations were performed and analyzed. The intra-observer variability was determined by calculating the DICE and Jaccard similarity coefficients. This provides segmentation similarity measures for the manual segmentation of the femur, tibia, femoral prosthesis component and tibial prosthesis component in low-field MRI data sets with a total knee prosthesis.

The DICE score is determined by the pixels present in both segmentations ($|A \cap B|$) and the pixels in each segmentation ($|A|$ and $|B|$). The DICE coefficient is calculated as follows:

$$DICE = \frac{2|A \cap B|}{|A| + |B|} \quad (10)$$

From this, the Jaccard score can be calculated as:

$$Jaccard = DICE / (2 - DICE) \quad (11)$$

While both the DICE and Jaccard coefficients describe the similarity of two segmentations based on the overlap of voxels in the segmentations, the main difference between the DICE and the Jaccard scores is the rate at which they grow. The DICE score will become larger faster than the Jaccard. which may provide more promising results than may actually be the case. A DICE or Jaccard alone does not say too much about the agreement between the datasets, while examining both simultaneously provides more insight in the similarity of the segmentation results.

5.1.4 Registration

During the registration step, the segmentations of the tibial prosthesis component and the femoral prosthesis component are required. The 3D models of the tibial and femoral prosthesis components, made with the optical scanner at the University of Twente, are registered to the segmentation models. The registration is performed in Mimics Innovation Suite 19 (Copyright Materialise 2017) using the global registration algorithm in the Stl registration option. Registration with the built-in registration algorithm for Mimics provides the remaining error as a least squares distance in *mm*. Here, for each femoral and tibial component registration, the remaining error is provided. It is not known exactly how the registration is performed, or how the error is calculated, however, the registration itself is a rigid registration method. As such, the remaining error does provide some insight in the accuracy of the registration.

5.2 Results

5.2.1 Patients

All patients in this study gave informed consent for the use of their low-field MRI data for this study. Included were seven patients without symptomatic total knee prosthesis and one with symptomatic total knee prosthesis. Age, sex and prosthesis age are shown in table 2. Prosthesis age is time between the date of the TKA procedure and the date of scanning the patient with the low-field MRI scanner. The average age of the patients included in this study is 67 years. There are 4 women and 3 men in the non-symptomatic total knee prosthesis group, with an average prosthesis age of 493 days. The patient with symptomatic total knee prosthesis, however, has a prosthesis age of more than 9 years.

Patient	Age(years)	Sex	Prosthesis Age(days)
101	62	F	Unknown
102	62	M	374
103	66	M	422
104	58	M	497
105	78	F	637
106	61	F	454
107	78	F	573
201	67	F	3387

Table 2: The age, sex and prosthesis age for the patients included in this study. Note the large difference in prosthesis age for patient 201(with symptomatic knee prosthesis) compared with the other patients).

The femoral component, tibial component and PE liner sizes of each individual patient is shown in table 3. This information is crucial for the registration step, as well as for the interpretation of the liner wear results. As can be seen from the table, the femoral components range from E to F, whereas the tibial component size range is from 3 to 7. The liner size is the same for all patients except patient 104.

Patient Number	Femoral Component Size	Tibial Component Size	Liner Size (mm)
Pt 101	E-R	3	10
Pt 102	G-L	6	10
Pt 103	F-R	5	14
Pt 104	G-L	7	10
Pt 105	F-R	4	10
Pt 106	E-R	3	10
Pt 107a	E-R	6	10
Pt. 201	F-R	4	10
Pt. 107b	E-R	6	10

Table 3: Size of the knee prosthesis components for the patients included in this study. Femur size is denoted as the component size followed by an L or R for left or right. This data is important for the registration step, as it is of crucial importance to register the correctly sized 3D models to the segmentation. Also, it provides insight in the liner wear measurements. The pre-registration models that are used for the ideally aligned measurement are of sizes: Femur E-R, Tibia 5, Liner 10.

5.2.2 MRI Data Artifacts

As stated in section 1.3, it is expected that the use of low-field MRI results in images with less artifacts than when using standard clinical MRI. As a result, the contours of the prosthesis components, both femoral and tibial, are expected to be more visible than with standard clinical MRI. The images in figure 15 show possible artifacts and difficulties with imaging a total knee prosthesis in a low-field MRI scanner.

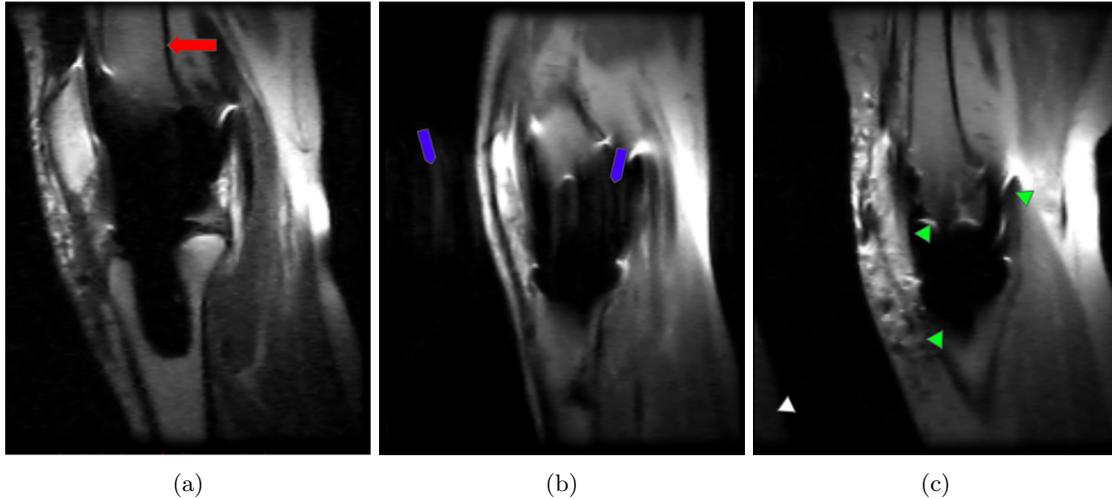


Figure 15: Several of the images retrieved from the low-field MRI scans performed on the patients. These images present a few of the major difficulties of imaging a total knee prosthesis with low-field MRI. a) shows a very short femur (red arrow), b) shows a movement artifact(blue arrows), and c) presents an example of the artifacts present in each data set(green arrowheads) as well as a wraparound artifact (white arrowhead). While these artifacts are also present in standard clinical MRI, the artifacts in the low-field MRI data was not expected to be this large.

5.2.3 Segmentation

3D models obtained from the manual segmentation using Mimics can be represented with meshes, containing vertices and face, for ease of use and straightforward processing. An example of a 3D model of the segmentation of a femur and tibia bone segmentation are shown in figure 16. A full segmentation model of all the segmentations, including component segmentations, is presented in figure 17. It should be noted that in both the femoral and tibial segmentation the models seem "sliced". This feature is more pronounced in the femoral coronal view where gaps in the model can be seen. This sliced appearance may be caused by the 4mm slice thickness.

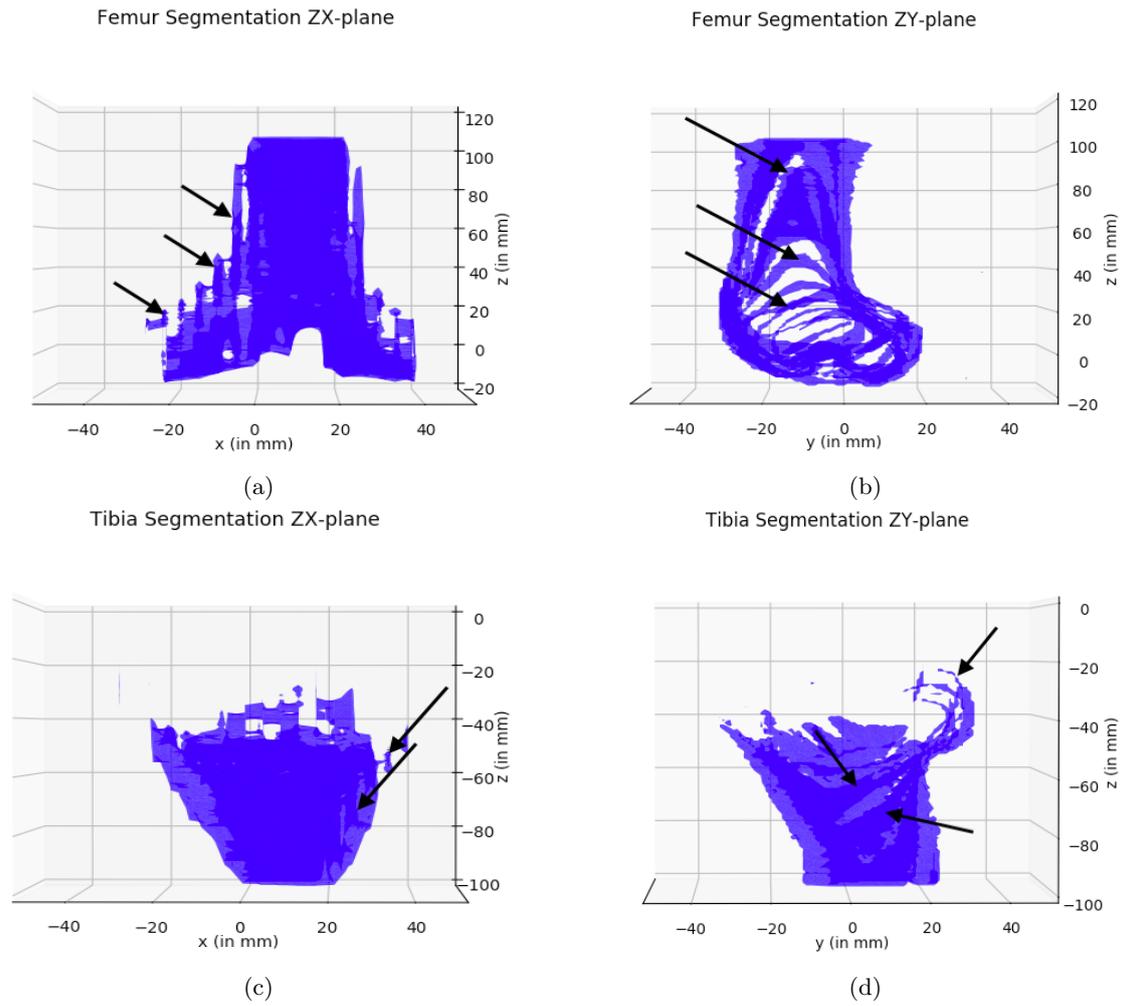


Figure 16: Examples of a segmentation. The 3D data is represented as a mesh, composed of faces and vertices. The blue represents the mesh of the segmentation. The black arrows show the "sliced" appearance of the segmentation.

Segmentation of MRI Data
Femur, Tibia and the Components

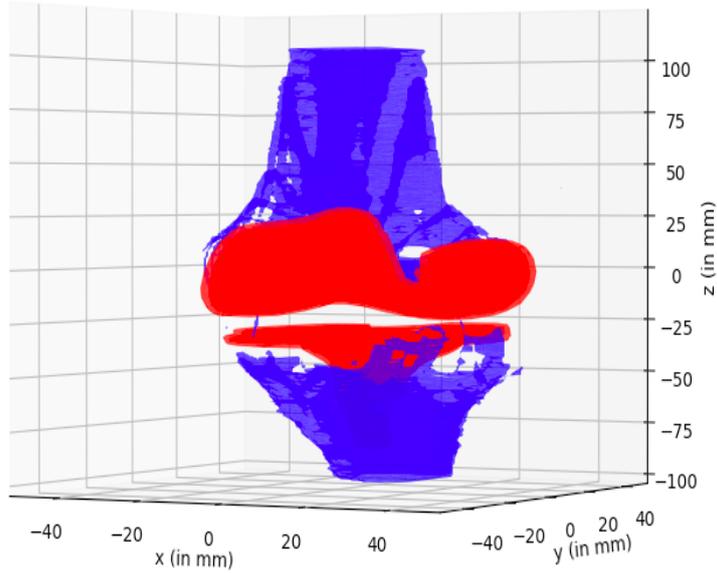


Figure 17: A complete segmentation of a single MRI dataset. All relevant bony and prosthesis structures have been segmented: femur and tibia (blue), and the femoral and tibial components (red).

The DICE and Jaccard indices calculated from the manual segmentations of the primary dataset are shown in table 4. As there are four segmented components, the femur, the tibia, the femoral prosthesis component, and the tibial prosthesis component, the DICE and Jaccard coefficients for the components are also presented. Note that the segmentation for the prosthesis components have higher similarity scores than the segmentation for the bones. Furthermore, the similarity between the second and third (2v3) segmentation is higher, suggesting a learning curve or prior information bias may be present.

		Seg1 v Seg2	Seg1 v Seg3	Seg2 v Seg3
Complete segmentation				
	DICE	0.63	0.57	0.76
	Jaccard	0.46	0.40	0.61
Bone Segmentations				
Femur				
	DICE	0.67	0.66	0.72
	Jaccard	0.51	0.49	0.56
Tibia				
	DICE	0.61	0.63	0.73
	Jaccard	0.44	0.46	0.57
Component Segmentations				
Femur				
	DICE	0.80	0.80	0.87
	Jaccard	0.67	0.67	0.76
Tibia				
	DICE	0.69	0.79	0.80
	Jaccard	0.53	0.65	0.67

Table 4: DICE and Jaccard scores of the three manual segmentations made with the primary low-field MRI dataset. A higher value indicates a better agreement between datasets. The component segmentations show a higher agreement between segmentations than the bone segmentations for all three segmentations. Seg1 is the first segmentation, Seg2 the second, and Seg3 the third segmentation.

5.2.4 Registration

Table 5 shows the results of the registration algorithm applied in Mimics to register the 3D models from the optical scanner to the segmentation models obtained from the previous segmentation step. The numbers presented are the remaining least squares error distance in *mm*, and a higher error indicates a less accurate registration, where the most accurate registration provides no remaining error.

	Femoral Component Registration (mm)	Tibial Component Registration (mm)
Pt 101	1.7127	2.3497
Pt 102	2.0617	2.3634
Pt 103	2.6073	2.5391
Pt 104	2.1047	1.8896
Pt 105	2.5613	2.5110
Pt 106	2.6772	2.2738
Pt 107a	2.0715	2.5625
Pt 201	2.3090	2.3224
Pt 107b	2.6361	2.1214

Table 5: Remaining registration error from the STL registration method in Mimics. The remaining error is determined as a least squares error distance in mm. The higher the remaining error, the less accurate the registration is. Here, the remaining error is similar for all the patients, with Pt. 106 having the highest femoral registration error and Pt. 107 has the highest tibial registration error.

5.3 Discussion

5.3.1 MRI Data Acquisition

As the results in figure 15 of the MRI data acquisition show, the data contains many artifacts. It was hypothesized that the use of low-field MRI would diminish the susceptibility artifacts caused by the presence of metals, such as that the shape of the metallic prosthesis components would be better visible than with a standard MRI exam. However, as can be seen from figure 15c, the susceptibility artifacts are still present, and it remains difficult to distinguish the individual components and identify the correct shape of the components. Standard MRI of a total knee prosthesis also shows significant artifacts (figure 18). The images from this study show similar artifacts shown in figure 10, indicating that the image quality with respect to metal artifacts of low-field MRI is not as good as expected.

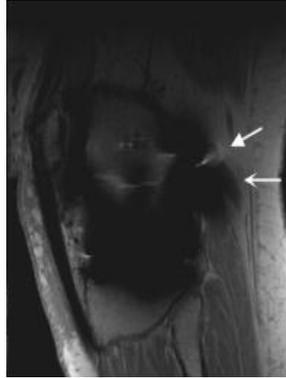


Figure 18: Sagittal MR images of a TKR in a healthy volunteer. Metal-induced artifact including distortion (wedge-shaped arrow) and signal loss (curved arrow) severely limit the diagnostic value of Fast Spin Echo (FSE) images. (image from Chen *et al.* [83])

Furthermore, the pixel spacing, or slice thickness, of 4.4mm between slices is very large, which increases the strength of the partial volume effect when compared to smaller voxels. This may be of importance when considering susceptibility artifacts. A smaller slice thickness means that the artifact does not provide as much of a pixel shift (Δr in equation 7) in the image. Therefore, decreasing the pixel spacing between slices, may provide more spatial resolution and could possibly decrease the size and intensity of the susceptibility artifacts present in the current images.

The use of promising magnetic artifact reduction sequences such as MAVRIC and SEMAC-VAT [84, 85], has been shown decrease metal induced susceptibility artifacts and increase the image quality. In this study, these sequences were not used as these specific sequences are not available on the low-field MRI at the University of Twente. Other sequences are available, such as X-MAR sequences, however, during the selection of the sequences that will be used for the imaging protocol, the X-MAR sequences were deemed of low quality compared to the images used currently. Considering the extent of the susceptibility artifacts in the data, the current images do not provide the artifact reduction that was expected from using low-field MRI. As such, a more artifact reducing sequence should be considered in the future if low-field MRI is to be used.

5.3.2 3D Scans

3D optical scans were obtained with an optical scanner at the University of Twente. These models are representations of the real-world prosthesis components. The component scans contain upwards of 15000 faces for a femoral or tibial component scan. After reduction of the faces, upwards of 7500 faces are still present. This is extremely large and partly causes the malalignment and liner wear measurements in which these models are used, to be fairly slow (several hours for 15000 faces, 45 minutes for 7500 faces).

As described in the above methods section, only one size of each prosthesis component was scanned, while the other size models were scaled from the scanned data. This does provide the correct medio-lateral and antero-posterior dimension, however, the scaling is uniform. The actual prosthesis components are not uniformly scaled from one size to another [86].

Also, the prosthesis components scanned with the optical scanner are not the same as those used

in the clinical practice of OCON. Ideally, the CAD-models from the manufacturer would provide more accurate registration results and possibly provide more accurate liner wear and malalignment results as well.

5.3.3 Segmentation

For the data in this study, a high score of 0.76 and 0.61 in the DICE and Jaccard scores, respectively, were found for the similarity of the complete segmentation of segmentation 2 and 3. These scores can be attributed to the lower scores found for the bone segmentation than for the component segmentation. As table 4 shows, the bone segmentations for both the tibia and the femur have a lower similarity than the component segmentations for all of the segmentations. This indicates that the bone is more difficult to segment than the component, while this would not necessarily be expected, given the presence of artifacts at the expected position of the components in the images.

The average similarity of the segmentations falls below the 0.70 threshold, above which is considered good agreement [87], for both the DICE and Jaccard scores, indicating the segmentation method is not very reproducible. This finding further supports studies showing manual segmentation has a high intra-observer variability [81, 82].

While manual segmentation remains the gold standard for segmentation methods [81, 82], studies have shown that the intra-observer variability is significantly lower for semi-automatic segmentation than for manual segmentation of distinct structures [88, 89]. Due to the remaining presence of artifacts in the low-field MRI data, accurate (semi-)automatic segmentation of the prosthesis components is difficult. The absence of defined contours of the prosthesis components and the bone segments, makes automatic algorithms based on thresholding or shape matching difficult to implement. As such, manual segmentation was performed for the segmentation task in this study. The use of (semi-)automatic segmentation may provide faster and more robust segmentation of the data, however image quality must improve. Metal artifact reduction sequences have shown to improve image quality of MRI exams of metallic objects [84, 85, 90]. Applying these sequences may provide a basis for the application of (semi-)automatic segmentation algorithms.

The segmentation results also show higher similarity coefficients of the component segmentations. This implies that the segmentation of the components is not necessarily the biggest issue with the method, whereas that was initially expected. The component segmentations even reach a good agreement or better (above the 0.70 threshold) for both the DICE and Jaccard coefficients for both the femoral and tibial component segmentations.

Other similarity coefficients, such as the Hausdorff distance were also considered, however, they were not expected to provide interesting information in addition to the DICE and Jaccard coefficients. The Hausdorff distance measure determines the closest distance of erroneous segmentation voxels from the ground truth segmentation. Here, a manual segmentation method was applied where singular outlying voxels can easily be spotted and removed from the segmentation during the manual segmentation step in Mimics. The Hausdorff distance was therefore not considered to provide interesting information about the similarity of the segmentations. [91]

5.3.4 Registration

The registration results are hard to interpret, because the registration algorithm used for the registration in Mimics is not known. However, it is suspected that the registration algorithm is some form of a point-cloud matching method. As the 3D scans that are used for this method contain a very high number of vertices compared to the segmentations, an error is expected to remain, as not every point will be exactly matched. However, there does not seem to be much variety in the remaining registration error, indicating the prior similarity results of the good agreement between component segmentations most likely hold true for the other data sets as well. This also implies that a possible error in the measurement does not result from varying registration results as these are fairly similar. Nonetheless, it is unclear if the results of the registration are good or bad.

6 Specific Analysis: Malalignment

This section will describe the materials and methods and results for the malalignment part of this study, as well as provide a measurement specific discussion of the method and results. The malalignment measurement is part of the specific analysis segment of this study.

6.1 Materials and Methods

Malalignment is measured in several steps. First, the anatomical axis or mid-shaft line of the bone is determined using a convex hull measurement. Then, the corresponding axis of the registered prosthesis is calculated. Lastly, the angle between the two axes is determined, which provides the relevant angles to determine the cFA, cTA, sFA, sTA, and TFA. The method described below is performed for both the femur and the tibia, for sagittal and coronal orientations.

Convex Hull Centroids

To determine the relevant angles as shown in figure 5, the anatomical axis of the bone should be calculated. To provide the anatomical axis, a similar approach to Stan *et al.*(2013) [39] and Miranda *et al.*(2010) [92] is taken, where the convex hull is used to calculate the central points of the bone and a line through the central points is determined. The convex hull is an algorithm to determine the smallest convex set of points that contains all the points in the respective slice. An analogy is shown in figure 19, where an "elastic band" is stretched around a set of nails/points. Then the band is let loose, the form and position of which then represents the convex hull of the set of points.

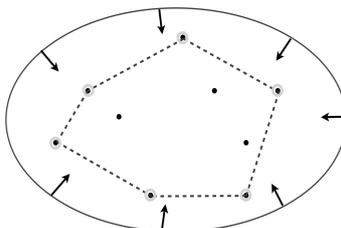
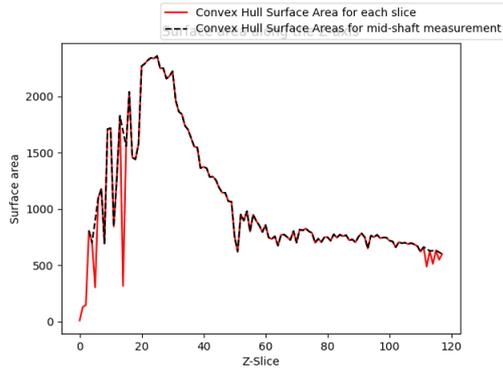


Figure 19: Example of the convex hull calculation. Here, the wide circle represents the stretched elastic band and the dotted line represents the resulting convex hull when the band is released. The points are random points in 2D space. (From [93])

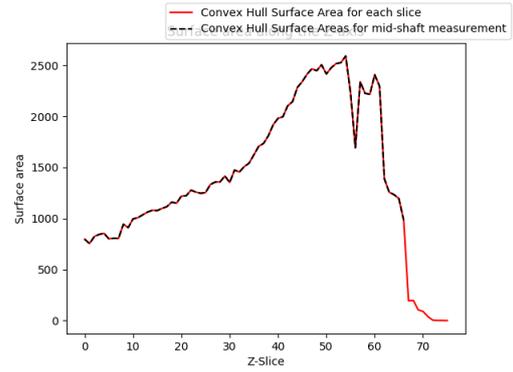
After the convex hull is determined, the respective surface area and centroids can be calculated. The surface area is an important parameter to determine which slices will be taken into account for further analysis. The centroids of the convex hull surfaces combine to create the set of centroids that will be used to approximate the anatomical axis of the bone.

Convex Hull Surface Area

The convex hull surface area is calculated using a shoelace algorithm [94] along the vertices of the hull. The resulting graphs shown in figure 24 show the surface area plotted against the slice number of the bone mesh along the global z-axis. Contrasting the method of Miranda *et al.*(2010) [92], instead of the upper $1/3^{rd}$ removed from the dataset, here the lower $1/4^{th}$ is discarded.



(a) Surface area of the convex hulls calculated along the femur.



(b) Surface area of the convex hulls calculated along the tibia.

Figure 20: Convex hull surface area for each slice along the z-axis of the segmentation. Furthermore, the black line shows the slices that are taken into account for determining the mid-shaft of the bone, *i.e.* up to $1/4^{th}$ of the maximum convex hull surface area.

This is done to account for outliers that can occur due to segmentation issues. Furthermore, during TKA surgery, a large part of the femur is removed, rejecting the assumption in Miranda *et al.*(2010) [92] that the distal part of the femur has the largest surface area by default. Hence, all data above $1/4^{th}$ of the maximum convex hull surface area is used for further analysis and measurements. Examples are shown in figure 16, where the blue mesh represents the bone segmentation.

Centroid Calculation

The convex hull centroid coordinates C_x and C_y are calculated by [95]:
$$C_x = \frac{1}{6A} \sum_{i=0}^{N-1} (x_i + x_{i+1})(x_i y_{i+1} - x_{i+1} y_i)$$

$$C_y = \frac{1}{6A} \sum_{i=0}^{N-1} (y_i + y_{i+1})(x_i y_{i+1} - x_{i+1} y_i)$$
 where A is the convex hull surface area, N is the amount of vertices contributing to the convex hull and x_i and y_i are the x and y coordinates of the i^{th} vertex of the convex hull. The z-coordinate of the convex hull centroid corresponds to the z-coordinate of the slice for which the convex hull is calculated. The collection of centroids is used for further analysis and approximation of the anatomical axis of the bone.

SVD Line fit

The next step is to calculate the anatomical axis of the bone segmentation. To provide an approximation of the anatomical axis, a straight line is fit to the data represented by the convex hull centroids calculated in the previous step. The line fit is calculated by a single value decomposition (SVD), which determines the main axes of a set of points and how well each axis describes the dataset. The axis describing the dataset the best is used to approximate the anatomical axis. One of the main things to note here is the points that seem to be outliers in the femoral centroids and tibial sagittal plane centroids images in figure 21. A correction for these outliers is applied to only include those points that fall within 2σ of the mean x- and y-coordinates of the set of centroids, resulting in the images in figure 22.

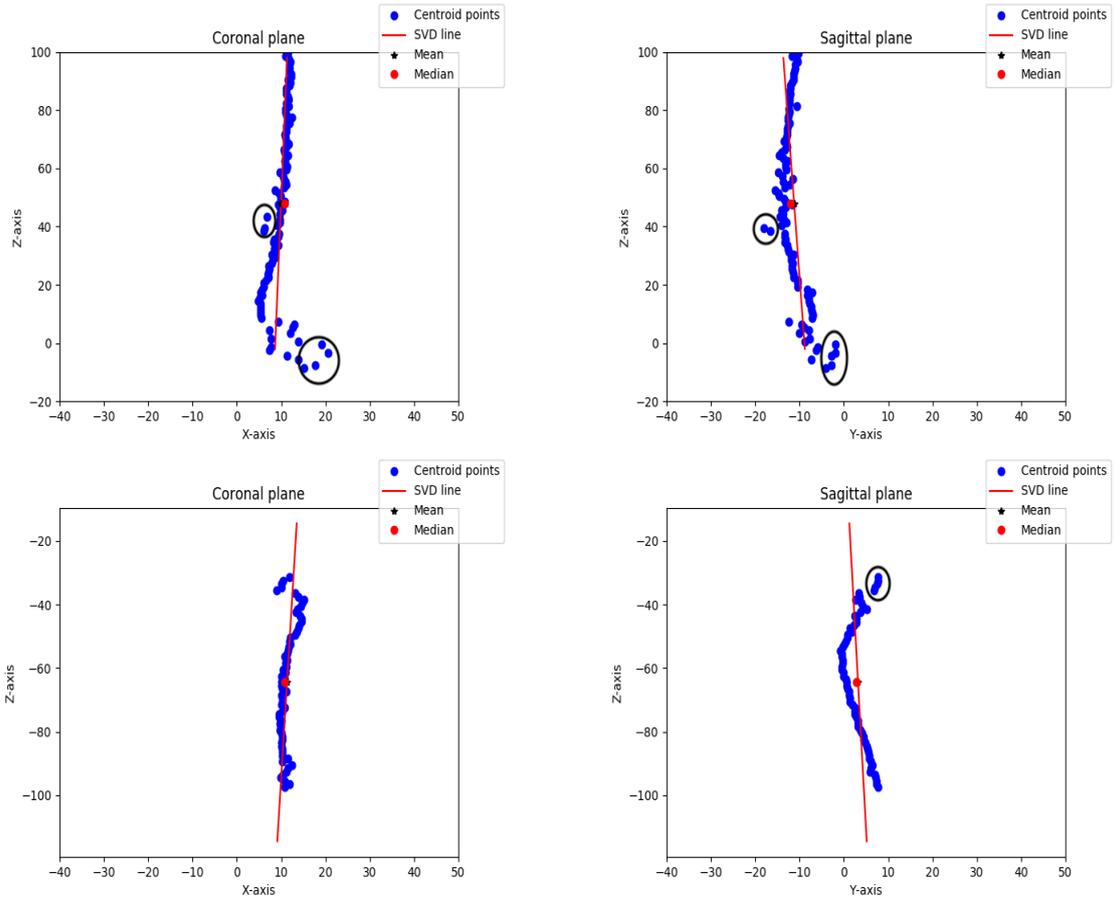


Figure 21: Uncorrected set of centroids of the z-slices of the femur and tibia segmentation in the coronal and sagittal planes. The line in each figure is the SVD fit of the data. The upper two images are femoral and the lower two are tibial centroid data fits. The black circles show the calculated centroids that seem to be outliers.

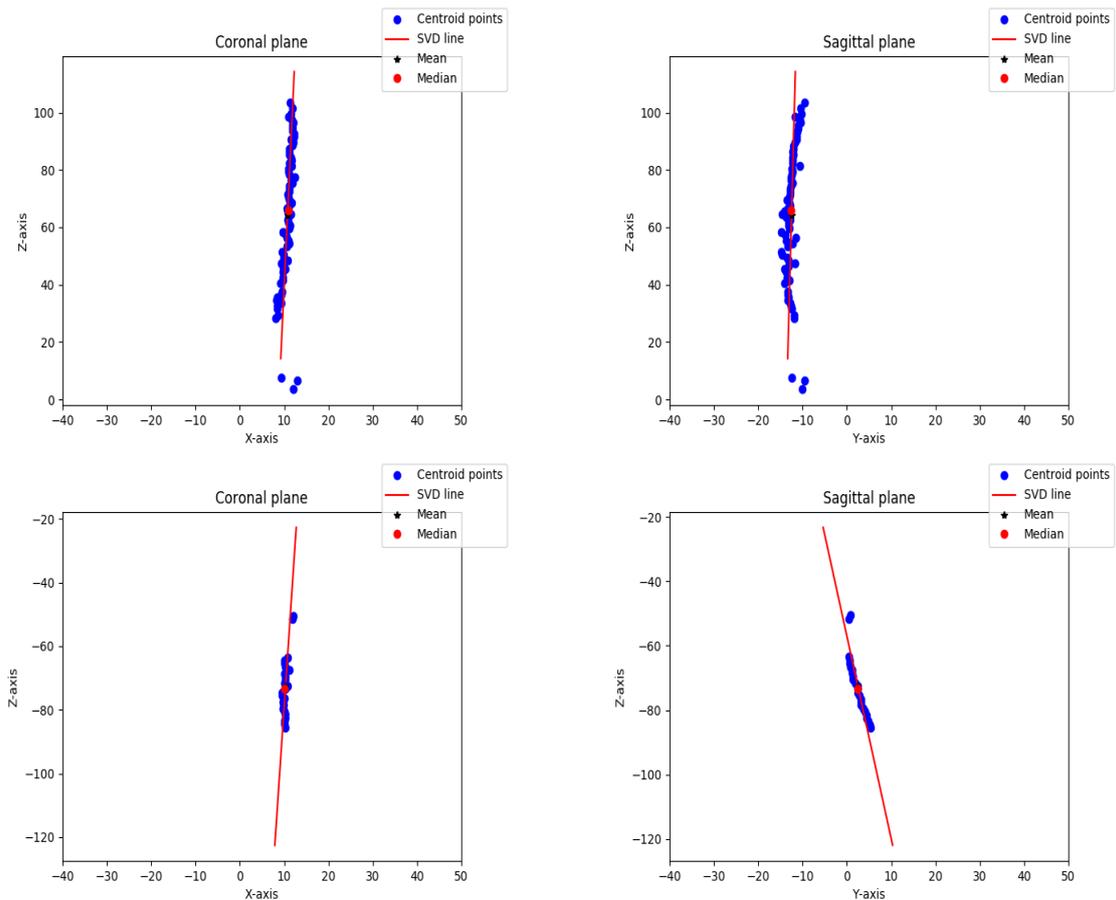


Figure 22: Corrected set of centroids of the z-slices of the femur and tibia segmentation in the coronal and sagittal planes. The line in each figure is the SVD fit of the data. The upper two images are femoral and the lower two are tibial centroid data fits. As can be seen, the outliers highlighted in the previous uncorrected data are no longer present.

Angle Measurements

The angles that are calculated as described below, are calculated with respect to the unit circle shown here. A negative angle, therefore, means that the angle between the calculated axis and the global Z-axis is in a clockwise direction. As such, an angle in counterclockwise direction is a positive angle.

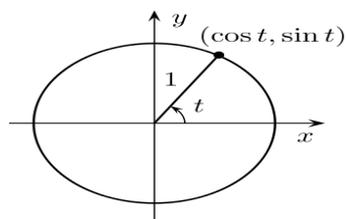


Figure 23: Unit circle indicating the direction of a positive angle [96].

The angle may be dependent on the cutoff of the maximum surface area and slice size chosen. The method presented above is applied to a cylinder which is rotated by a pre-determined angle. The cylinder is skewed to mimic the shape of the femur, a large diameter at the bottom and a small diameter at the top. The results in the figures below show the effect of the choice of either $1/4^{th}$ or $1/3^{rd}$ of the maximum surface area. Other convex hull surface area cutoff values were also tested, however, those did not provide similar results to those shown here.

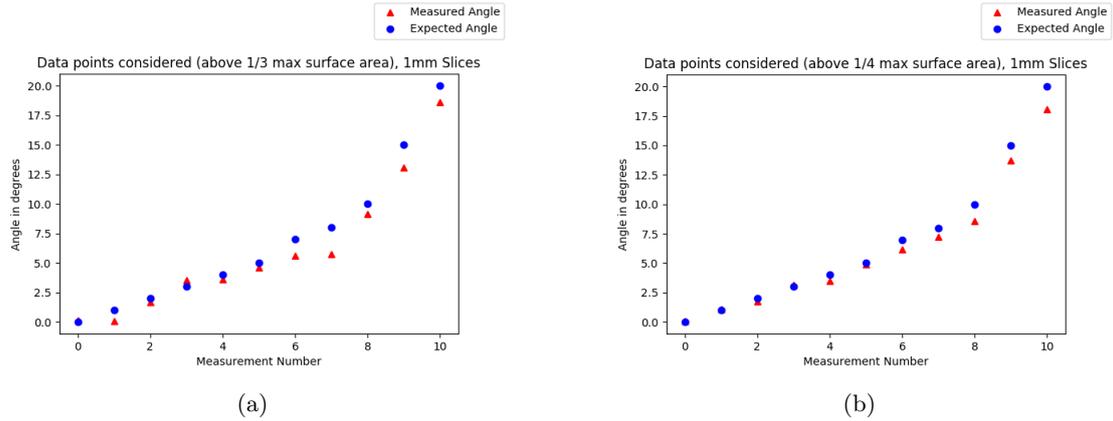


Figure 24: Results of the angle calculation of a testing cylinder when varying the convex hull surface area cutoff value. The cylinder was placed at a set angle (blue) in the ZX-plane, then the central axis was determined and the angle in the ZX-plane was calculated (red).

Due to the MRI scanning method, it is not expected that the angles measured in this study will not be very large, therefore, a more accurate measurement for the lower angles is important. As such, the $1/4^{th}$ convex hull surface area cutoff value is used for the analysis.

Bone Angle

The angles of the bone with respect to the global z-axis are calculated by determining the projection of the line provided by the SVD linefit on the zx- and zy-planes. From here, the inherent calculation using the dot product results in the angle between two lines:

$$\theta = |A||B|\cos^{-1} \quad (12)$$

where A is the projection of the linefit on the plane, B is the global z-axis and Φ is the angle between A and B in radians. The angles that are calculated for the sagittal and coronal planes are $\theta_{sagbone}$ and $\theta_{corbone}$ respectively.

Component Angle

The prosthesis component is aligned to the global coordinate system, as described previously in the methods section, *i.e.* the x-, y-, and z-axes of the prosthesis component LCS are the same as the GCS. During the registration step, the prosthesis component is registered to the 3D segmentation in Mimics Innovation Suite 19. This provides a 4x4 registration matrix \mathbf{T} , that includes a translation (t) and a rotation (R). The registered prosthesis component axes are calculated by applying the registration matrix \mathbf{T} obtained in the registration step, to the GCS. In a similar manner as above, the angles between the z-axis of the registered prosthesis component and the x- and y- axes of the GCS ($\theta_{corprot}$ and $\theta_{sagprot}$, respectively) are calculated.

Malalignment Angles

From the angles that are calculated from the above steps, the cFA, cTA, sFA, sTA and TFA can be determined. Up to this point, all the angles that are calculated, are with respect to the GCS. To calculate the clinically relevant angles, the following calculations are performed. The cFA is calculated by: $90 + / - (\theta_{corprot} - \theta_{corbone})$, depending on whether the data is from the left or right knee. As the cFA is always determined on the medial side, whether the left or right knee is considered is important. A similar approach is taken for the sFA *i.e.*, $\theta_{sagprot} - \theta_{sagbone}$. The whole process is then repeated for the tibia and tibial component, resulting in the cTA and sTA. The TFA is then calculated by: $\theta_{corboneFemur} + \theta_{corboneTibia}$. Furthermore, the alignment angle, or balance, between the components is determined. This angle, as shown by Stan *et al.*(2012) [39] is key to reduce the stress on the liner and tibia. This angle, referred to here as the component balance angle (CBA), is determined by: $cFA + cTA$. The resulting clinical malalignment angles are cFA, sFA, cTA and sTA.

Alignment Angles



Figure 25: The angles cFA, sFA, cTA and sTA will be calculated as shown in the above image. Here, the sFA parameter has a positive angle. In this figure, the CBA and TFA are not shown. (Image adjusted from [38])

Expected Results

From Stan *et al.* [39], the recommendation is to balance the components to minimize the forces on the tibial plate and tibia. A component balance angle (CBA) of 180° is recommended by Stan *et al.*, however, other studies recommend the ideal coronal alignment is between 3° varus and 7° valgus [32–34,97]. From literature, it is shown the component balance angle (CBA), should be in the range of 177 – 187° . In this study, the range of 177 – 187° was used for the CBA as a reference range. The other reference value ranges are determined from previously mentioned literature [32–34,97] and are as follows: the ideal alignment should be between 3° varus and 7° valgus. For a cTA of around 87° , that means the cFA should be between 84 and 94° , the sTA should ideally be 90° and the sFA is ideally 0° . The TFA should be approximately 175 – 176° .

6.2 Results

For the malalignment measurement, a line was required to be fit to the centroid data. In this study, an SVD linefit was used for determining the straight line through the centroids. A least squares algorithm was also considered, however, it was deemed that the resulting line did not match the data as well as the SVD line fit did considering the results shown in figure 26. From these figures it was considered that the SVD line fit performed superior to the least squares algorithm and was therefore used for the calculation.

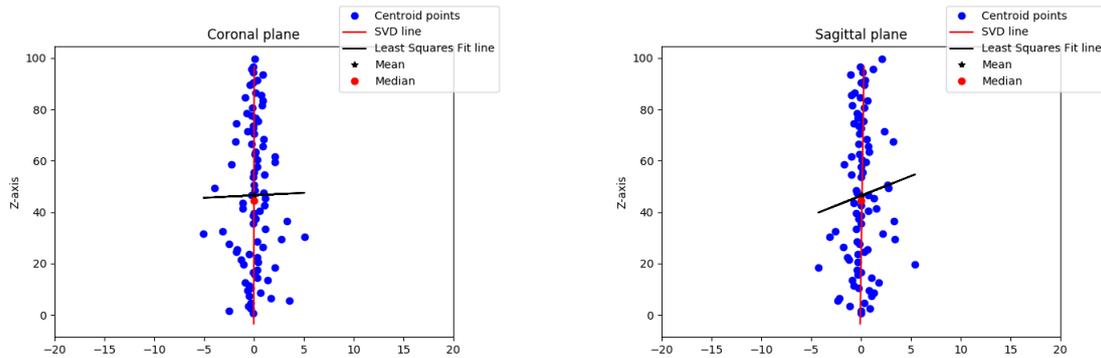


Figure 26: The above figures show the line fits of femoral centroids made with the SVD and least squares methods. As can be seen, the SVD fits the data better than the least squares estimation.

The results of the malalignment measurement for each dataset are given in table 6. Figure 27 provides a reference for the calculated angles.

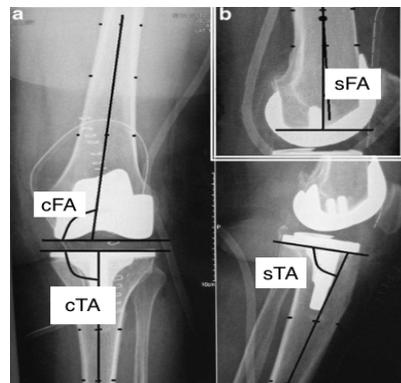


Figure 27: Malalignment angles calculated in the malalignment measurement. The cFA, cTA, sFA and sTA are shown in this image. It should be noted, in this figure, the sFA parameter is positive. The TFA is not shown but is calculated as the angle between the tibial and femoral anatomical axes. Furthermore, the CBA is also not shown here, however is simply calculated as: $cFA + cTA$. (Image adjusted from [98])

Table 6 presents the results of the malalignment calculations. It can be noted that the most of the values do approximate the expected clinically relevant reference values. However, the values for the cTA seem to be on the higher end for almost all the data sets, while the cFA is out of the expected range for pt. 106 only. The sTA values are mostly lower than the expected value of 90° , however only by a maximum of 10° . Further analysis of the data is shown in the boxplots in figures 28, 29 and 30.

	cFA(°)	cTA(°)	sFA(°)	sTA(°)	CBA(°)	TFA(°)
Clinical Range	84-94	87	0	90	176	177-187
Pt. 101	88.41	101.63	-5.32	84.69	190.04	176.29
Pt. 102	87.56	101.96	-12.77	92.21	189.52	180.09
Pt. 103	88.78	96.39	-7.60	80.59	174.83	176.82
Pt. 104	94.50	91.67	-2.66	92.96	186.17	192.19
Pt. 105	94.21	87.66	-5.65	87.15	178.13	167.11
Pt. 106	89.72	97.41	-12.87	87.80	172.87	179.96
Pt. 107a	93.35	90.81	3.11	89.16	175.84	174.21
Pt. 201	83.64	99.17	-7.50	88.93	177.19	183.26
Pt. 107b	103.04	83.31	-0.08	84.02	173.65	156.01

Table 6: Calculated values of the malalignment angles. cFA = coronal femoral angle, cTA = coronal tibial angle, sFA = sagittal femoral angle, sTA = sagittal tibial angle, TFA = tibiofemoral angle, CBA = component balance angle. The bold values are within the clinical reference range (or within 5 degrees of a single clinical reference value).

The figure28 shows the boxplots of the healthy patient data with the scanner in vertical (standing) position (Pt. 101-107a). It should be noted that the whiskers of the boxplot reach to 5% and 95% of the dataset, whereas the boundaries of the box are the second and third quartile of the data. Also, the clinically expected values for the cFA, sFA, TFA and CBA (green values in the plot) fall (at least partly), within the 95% CI range of the data. For the cTA and sTA, the reference values fall outside of the 95%CI of the values found in this study.

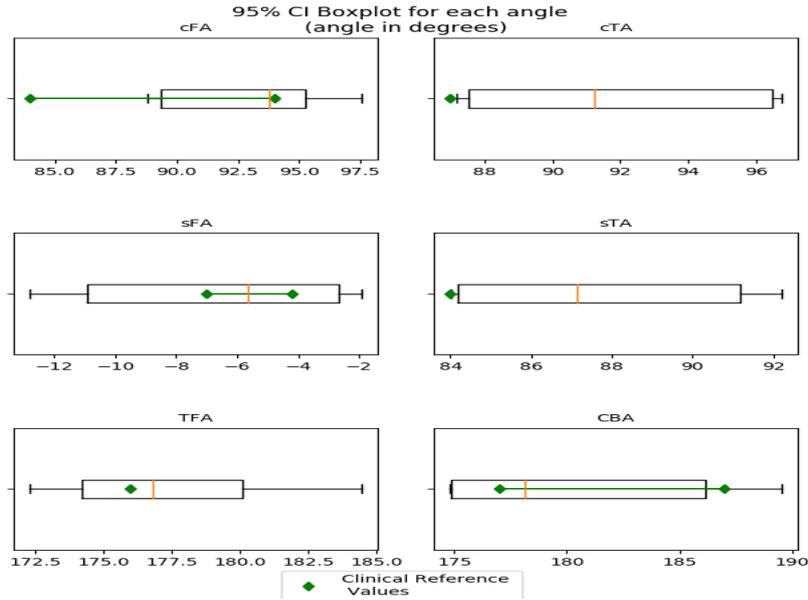


Figure 28: Boxplots for the values calculated for the malalignment angles as well as the clinical reference values. As can be seen, the clinical reference values fall within the calculated ranges of this study, except for the cTA and sTA. Also, the cFA reference range partly coincides with the values in this study. Here, the green data in the plot are the reference values described in the literature.

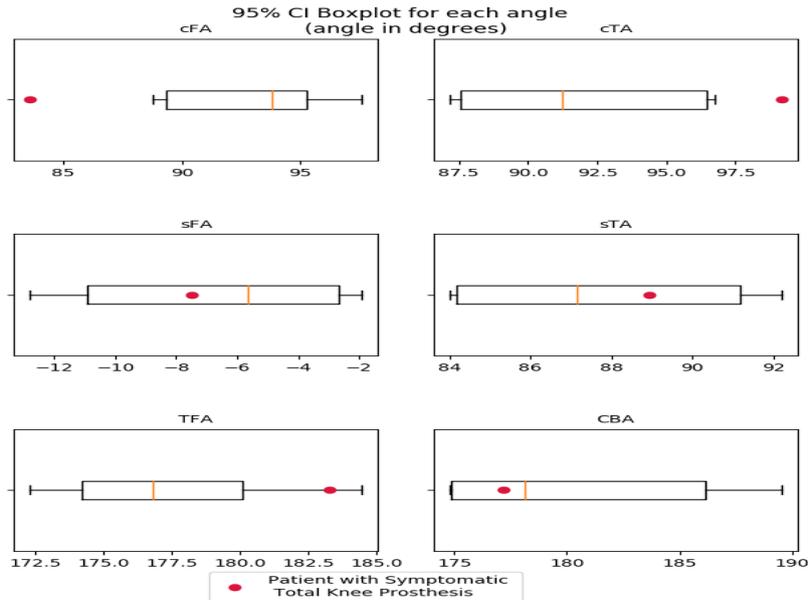


Figure 29: Boxplots with 95% CI of the malalignment angles calculated in the malalignment measurement. The red data indicates the values found for the patient with symptomatic total knee prosthesis.

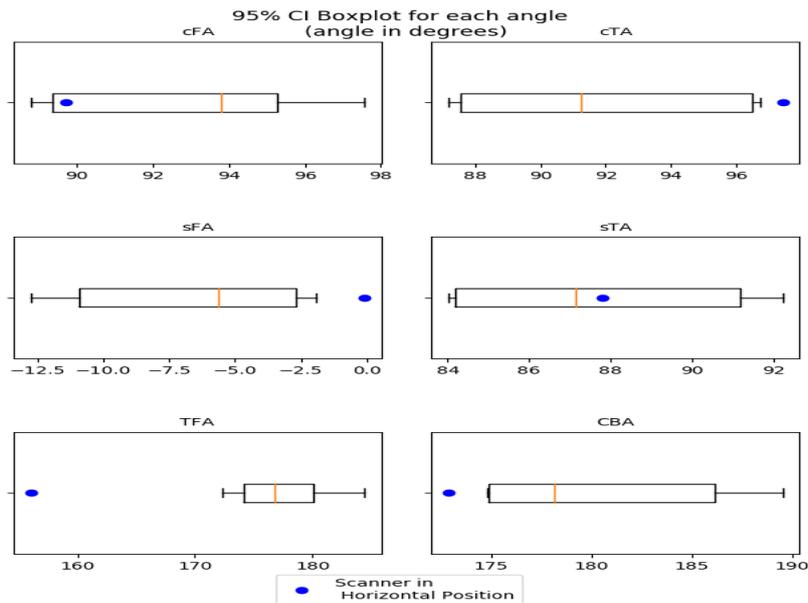


Figure 30: Boxplots with 95% CI of the malalignment angles calculated in the malalignment measurement. The blue data indicates the values found for the data scanned with the patient in supine position (scanner in horizontal position).

The figures (29 and 30) show the results of the measurements made with the scanner in horizontal position (30), and with a patient with symptomatic total knee prosthesis(29). Figure 29 shows that the values calculated for the patient with symptomatic total knee prosthesis are within the 95% CI ranges of the sFA, sTA, TFA and CBA found in this study. For the cFA and cTA the values calculated for the symptomatic patient fall outside of the 95%CI range as well as the clinical reference value ranges. It should be noted however, that except for the cFA, the results for the symptomatic total knee prosthesis patient are within the ranges of the healthy controls. Figure 30 shows the results of the values calculated with the scanner in horizontal position. Here, the values for the cFA and sTA only, correspond to the 95% CI of the data. Furthermore, the cFA and sTA fall outside the clinical reference value ranges. The cFA for Pt. 107b is 9.04° larger

than the clinical reference value and 8.54° larger than the highest other cFA value in the results of this study. Also, the TFA of Pt. 107b is 20.99° smaller than the clinical reference value range and 11.10° smaller than the lowest other TFA value in the results of this study. Considering the large difference between the range of values found for the upright scanning values and the cFA and TFA results for Pt.107b, these two parameters show an interesting difference between upright and supine scanning positions may exist. These values are also outside the clinical reference value ranges, which again suggest a difference between upright and supine image data.

6.3 Discussion

As indicated earlier, the cTA and sTA angles are higher and lower, respectively, than the values that have been indicated in literature [99]. However, other research shows directly post-operative angles of 87 and 84 for cTA and sTA, respectively [98], which more closely resemble the values found in this study. Furthermore, Kilincoglu *et al.* also reported sFA values of 6.6 ± 2.4 , whereas other studies do not disclose or determine these values. These reported values are measured directly post-operatively, whereas in this study the average age of the prosthesis components is 493 days. A reason for the differences in values may be due to possible micro-motion that takes place over time.

The differences in the values found with the method presented in this study compared to the clinical reference values, imply that the method is currently not accurate enough to provide reliable measurements for clinical application. As stated above however, the differences in these values may be due to the prosthesis age of the patients in this study, which may explain the large variance in the results shown in figure 28. However, from the same figure, as well as table 6 it can also be seen that a large part of the values do correspond with the clinically relevant reference values. This implies that the method does have the capacity to determine the malalignment values for the clinical ranges. As such the method does show promise for future applications.

Another finding is that the results of the patient with remaining complaints of the total knee prosthesis, pt. 201, are not very different from the other results. When inspecting the results, it can be seen that the results for pt. 201 only differ from the non-symptomatic total knee prosthesis patients results in the calculated cFA. This lack of dissimilarity between the results may be caused by the inaccuracies of the segmentation that subsequently influence the registration and the malalignment measurement. However, it could also be that the pathology of the patient is not caused by the malalignment of the prosthesis components and complaints originate from other pathology.

Lastly, it is interesting to note that the femoral measurement for pt. 102, does not differ in the coronal direction but does in the sagittal direction. Due to the short femur in the image data of pt. 102, the malalignment measurement for the femoral variables is expected to possibly differ from the other measurements in the study. However, this does not seem to be the case for the coronal femoral angle. As such, it seems that the measurement is robust in the coronal plane. This may imply that the minimum length of the femur required in the image for adequate anatomical axis approximation may not be 5.5mm as suggested by Miranda *et al.*(2010) [92] for the coronal plane.

For the malalignment measurement, only the data of slices with a convex hull surface area of more than $1/4^{th}$ of the maximum convex hull surface area were taken into account for further measurement. This was done to reduce outliers in the data. As discussed previously, outliers are likely to be present in the data due to the presence of susceptibility artifacts in the MRI data which hinder the manual segmentation. The result is that at the intersection between the prosthesis component and the bone, the segmentation may not be very good and result in lower or higher surface areas than expected. An example is shown in figure 24, where the left image describes the convex hull surface area from the bottom to the top of the femoral segmentation data. The jagged rise of the curve in figure 20a shows that the convex hull measurement does not result in a fluid increase in the convex hull surface area up to the maximum. From Miranda *et al.*(2010) [92], however, a more fluid increase of the convex hull surface area up to the maximum is expected. The jagged curve may be the result of earlier segmentation errors or as a result of the partial volume effect. It was expected that these errors are more frequent at the lower convex hull surface areas, therefore, unlike Miranda *et al.*(2010) [92], here the lower $1/4^{th}$ of the surface area slices are disregarded.

This, however, does leave possible errors at the larger end of the bone structure. An example occurs in Pt. 102, where only a small femur is depicted in the MRI data, see figure 15. The resulting alignment angles are determined by the largest part of the femur segmentation. In Miranda *et al.*(2010) [92], the largest part of the femur, i.e. the femoral condyles, are not considered for further imaging, because the anatomical femoral condyles influence the measurement of the anatomical axis when considering the convex hull approach. In the future, filtering the erroneous slices with a too large surface area should also be considered. However, enough data must be maintained to perform the measurement, which is the reasoning for not attempting to filter too large slices in this study as well.

7 Specific Analysis: Liner Wear

This section will describe the materials and methods and results for the liner wear part of this study, as well as provide a measurement specific discussion of the method and results. The liner wear measurement is part of the specific analysis segment of this study.

7.1 Materials and Methods

Tibial Surface Grid Calculation

To measure the thickness of the PE liner of the prosthesis, it is assumed the liner is the only thing between the prosthesis components [15,100]. Thus, the distance between the tibial and femoral components is the thickness of the liner. To calculate the distance from the surface of the tibial component that surface must first be defined. The tibial component mesh is split in various bins of vertices by dividing the surface in 1mm slices. As the tibial component surface is a flat structure, it can be assumed that in the pre-registration step the slice containing the surface of the tibial component will have the most vertices. This slice then provides the z-component of the tibial component surface in GCS coordinates.

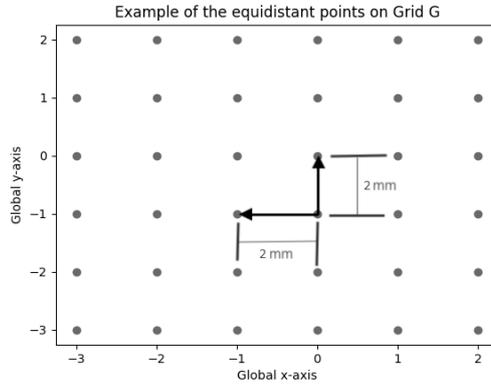


Figure 31: Several points of grid G . The distance between the points in each perpendicular direction is 2mm.

After the position of the surface is determined, a grid consisting of evenly spaced points is defined on the tibial surface. Each point on the grid is spaced a set distance from the next in each direction as shown in figure 31. The grid is sized by the maximum and minimum x-, and y-values of the scanned femoral and tibial components meshes. The defined grid is now a set of points G , on the surface of the pre-registration tibial component mesh. After applying the registration matrix for the tibial component obtained from the registration step to G , the grid will be oriented along the surface of the registered tibial component mesh, giving the new grid G_r . This step is illustrated in figure .

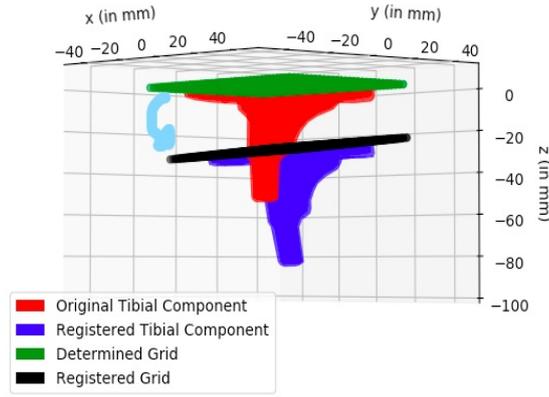


Figure 32: Calculation of grid G of the original tibial component. From the registration matrix the new position of the tibial component as well as the new position of the grid is determined, providing the new grid G_r . The blue arrow indicates that the registration matrix is applied to the green grid and red component.

Shortest Distance of Point to Line

The shortest distance between a point P and a line l_1 is the perpendicular line l_2 from point P to line l_1 (see figure 33). The points representing grid G , are the initial starting points of the perpendicular line in figure 33. The orientation of G is such that the global z -axis is its normal, n . By providing, for each point on G , a second point along the global z -axis above G , a ray can be defined by:

$$ray = P0 + k \cdot P1 \quad (13)$$

where $P0$ is the starting point of the ray on grid G and

$$P1 = P0 + n \quad (14)$$

with n the unit vector normal of grid G . To obtain the positions and orientations of ray on the registered tibial component, the registration matrix resulting from the registration step is applied to G , resulting in G_r . From this multiplication with the registration matrix, the ray vectors also rotate and translate and remain perpendicular to the defined grid.

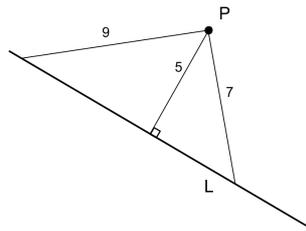


Figure 33: Example showing the concept of the perpendicular line from a point to a line. Here, three lines are drawn from point P to line L . The minimum distance from P to L is the perpendicular line (5). (from [101])

Line Plane Intersection

After the ray vectors have been defined, the last component of the measurement is the point P to which the distance should be measured. To calculate points P , the intersections of the ray vectors with the registered femoral component mesh need to be determined. The measurement of the intersection between a line and a plane is performed by calculating the normal of the plane and determining the dot product with the line. Considering a line described by $P0 + v$, where v is the direction of the line in the form of a unit vector and $P0$ is the 3-dimensional starting point of the line. The normal of the plane is calculated by considering the three points of the face, $p1, p2$ and

p_3 , each of which are 3-dimensional, and determining the output of the following equation

$$Face_{normal} = a_2b_3 - a_3b_2 - (a_1b_3 - a_3b_1)a_1b_2 - a_2b_1 \quad (15)$$

where $a = p_2 - p_1$ and $b = p_3 - p_1$. To calculate the intersection of the line and the face, an arbitrary point on the face is used, in this case the center of the face. From here, to calculate the position of the intersection, the equation $P_0 + k * v$ must be satisfied for the parameter k , a scalar number determining the length of the line. Calculating the parameter k is achieved by again applying the dot product

$$k = k_{numerator}/k_{denominator} \quad (16)$$

where $k_{numerator} = (p_1 - P_0) \cdot normal$
 $k_{denominator} = v \cdot normal$ Entering k into the ray equation: $P_0 + k * v$ provides the 3-dimensional position of the intersection between the line and the face. The intersection calculation is performed for each *ray* vector on the grid G_r and for each face separately. Performing this calculation will result in the same amount of intersections for each ray as there are faces in the mesh. Many of these intersections, however, are not relevant, as they are not within the boundary of the mesh, which is determined by the faces of the mesh. To determine whether the intersection is on a face, the face boundaries are used to determine a valid intersection. If the intersection falls within the face boundaries, an intersection is deemed valid, otherwise it is not.

The results of the liner wear measurement are provided as a heatmap, where the measured length is set out against the position of the respective measurement. A higher measured length corresponds to a darker color, and a smaller distance corresponds to a lighter color (see figure 34).

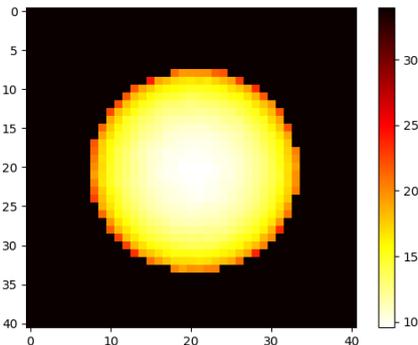


Figure 34: Presentation method of the liner wear measurement. Here, the result of a testing data set is shown where the distance from a 40mm x 40mm grid to a spherical object 10mm above the grid was calculated.

The thickness of the liner at the articulating surface is shown in table 7 for the range of liner sizes available. This information is crucial for the analysis of the measurement results. The articulating surface is the surface between the tibial plate and the femoral condyle(s) and represents the expected lowest value of the measurement for patients without liner wear.

Liner Size (mm)	10	12	14	17	20
Art. Surface Thickness(mm)	6	8	10	14	16

Table 7: Sizes of the various available liners, together with the thickness of the articulating surface for each size.

To increase the speed of the measurement, the distance between the points on the grid was changed from 1mm to 2mm. This decreases the amount of points on the grid to iterate through from 1640 to 410. This increases the speed four-fold as well. The decrease in points also results in a decrease in precision, however, there does not seem to be any visual difference between the results of the

1mm and 2mm measurements. The high increase in distance near the posterior edge of the measurement can be seen in both measurements as well as the 'v'-shape at the anterior side, however, increasing the distance between the points further may remove these important details from the result. For this study, the 2mm distance between points will be used.

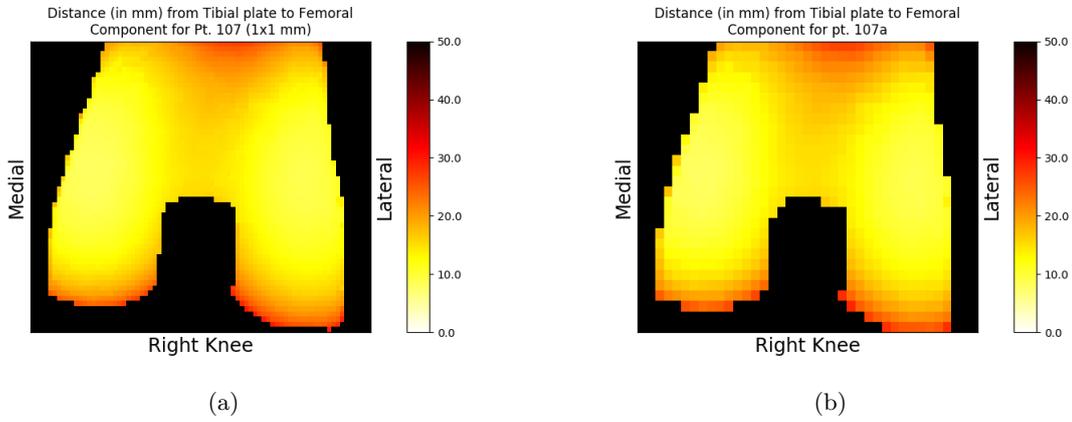


Figure 35: Measurement result for patient 107a with 1mm distance between the points a), and the measurement results for patient 107a with 2mm distance between the points b).

(a)

To increase the speed of the measurement further, the femoral mesh is halved. Each face of the femoral component mesh will provide a measurement result. This means that the top of the mesh also provides a result, which is not of interest for the liner wear. Also, it is not expected that a face in the upper half of the mesh will provide a small enough distance to be of use in this measurement. By removing the top half of the mesh from the data before the measurement, this will increase the speed of the measurement by a factor of almost two.

7.2 Results

Liner Wear Measurement Results

The measurement of the liner wear is represented as a distance heatmap. The result shown in figure 37 is the result from a liner wear measurement with the tibial and femoral components ideally aligned, i.e. directly above each other and without flexion/extension and varus/valgus. Here a right femoral component of size E, tibial plate of size 5 and a liner of size 10mm (articulating surface 6mm) are used. The darker the color, the higher the distance from tibial component surface to the femoral component.

	Mean(mm)	SD(mm)	Minimum(mm)
Ideal Alignment	11.31	5.18	4.28
Pt. 101	39.15	5.33	32.29
Pt. 102	9.80	4.34	3.45
Pt. 103	14.47	6.75	4.98
Pt. 104	8.21	4.73	0.58
Pt. 105	11.56	4.56	5.34
Pt. 106	11.20	4.05	5.33
Pt. 107a	14.28	4.42	7.92
Pt. 201	15.47	5.97	4.94
Pt. 107b	18.42	4.85	11.06

Table 8: Calculated mean, standard deviation and minimum values of the liner wear measurement for all patients in the study. All values are in *mm*. Note the very large mean and minimum of Pt. 101 compared to the other measurements.

Note that this initial measurement shows that the distance is expected to be lowest in the center, at the articulating surfaces, and highest near the anterior-posterior edges of the component (Ideal Alignment results in table 8). Furthermore, it must be noted that the measurement does not represent the shape of the liner. The actual shape of the liner is shown as an overlay in blue/green in figure 37. The liner wear measurement represents the distance from the tibial plate to the femoral component. At the grid points where the measurement does not determine a distance, the value is set to very high and is represented as black in the image. Due to the shape of the femoral prosthesis component, the measurement seems to show a hole in the center bottom of the heatmap, between the posterior edges of the condyles of the femoral prosthesis component. The results were analyzed with respect to two aspects, the visual appearance and their mean and standard deviation.

In the visual appearance, the shape of the measurement should be similar to the expected result shown in the figure below. The "v"-shape at the gradient from high to low distances at the anterior should be present, as well as the pronounced notch between the condyles. Furthermore, the higher distances should be at the anterior and posterior edges of the measurement, whereas the lower distances are located more centrally in the condyles. Note also the slightly larger area of seemingly lower distance at the lateral condyle in figure 37 compared to the medial condyle. These features are typically expected for the measurement.

The mean, standard deviation and minimum value provide a means to more quantitatively analyze the measurement. It is expected that the standard deviation of the measurements is similar to the ideally aligned results, however the mean and minimum may differ. If different, the mean and minimum values should differ from the ideally aligned result by a similar amount. For the ideally aligned measurement with femoral component size E, tibial plate size 6 and liner size 10, the following values were found: mean = 11.31mm, standard deviation = 5.18mm, and minimum value = 4.28mm.

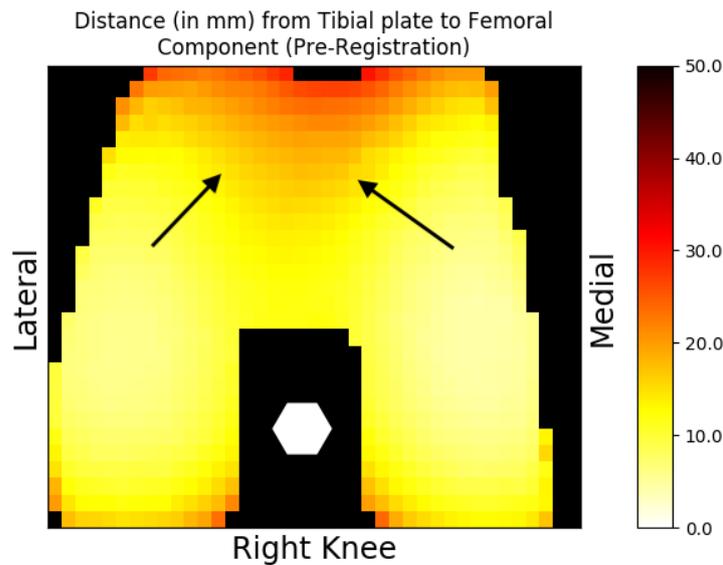


Figure 37: Result of the liner wear measurement when the femoral and tibial components are ideally aligned. Here a femoral component of size E, a tibial plate of size 6 and a liner of size 10mm are used for the measurement. This result will be referred to in the remainder of the report as the ideal alignment result. The black arrows point to the "v-shape" showing the increasing gradient toward the anterior edge of the measurement. The white hexagon shows the "hole" between the femoral condyles where no distance is measured.

The quantitative results for the liner wear measurements for patient 101 to 107 are shown in table 8. The mean, standard deviation (sd) and minimum value for each measurement is given, for the patients as well as the values for the ideally aligned measurement. Note the large values for Pt. 101 for the mean and minimum result of the measurement, when compared to the other results. Also, the resulting mean and minimum values for pt. 102 and pt. 104 are lower than the other values. These are also the only patients with a left-sided total knee prosthesis.

Figure 38 shows the results of the liner wear measurements for patient 102, 103, 104 and 106. The top left image, the result for patient 102, shows the result for the liner wear measurement in the left knee of the patient. The result seems like it is cropped, cut off at the anterior and medial sides of the measurement. This indicates that the femoral component is not aligned with the tibial component in the axial plane. There seems to be rotation and translation of the femoral component compared to the tibial component. Secondly, the "v"-shape at the gradient between high and low distances near the anterior part of the measurement in figure 37 appears more pronounced for the ideally aligned measurement, than for the result for patient 102. This can be attributed to the lower mean and sd of the results for patient 102 compared to the ideally aligned measurement. Lastly, due to the prosthesis being on a left knee in patient 102 instead of a right knee as in the ideally aligned measurement, the higher amount of lower values is still at the lateral condyle, however, the lateral and medial sides are switched compared to the ideally aligned result shown above.

For the result of patient 104, a similar result to patient 102 is obtained, however, it appears that only a rotation has occurred. Furthermore, the lateral condyle in this patient shows a far stronger difference with the medial condyle than the other results. This is also reflected in the large component balance angle value of 186.17° from the malalignment measurement.

The result for patient 201 shows a similar result to patient 102, however, appears to be mirrored with respect to the antero-posterior plane. Also, it shows a higher mean value $15.47mm$, while the standard deviation $5.97mm$ and minimum value $4.94mm$ are similar to the expected result.

The measurement results for patient 106 shows the result obtained for most of the other measurements as well. Here, the result closely resembles the expected result shown in figure 37, visually as well as quantitatively.

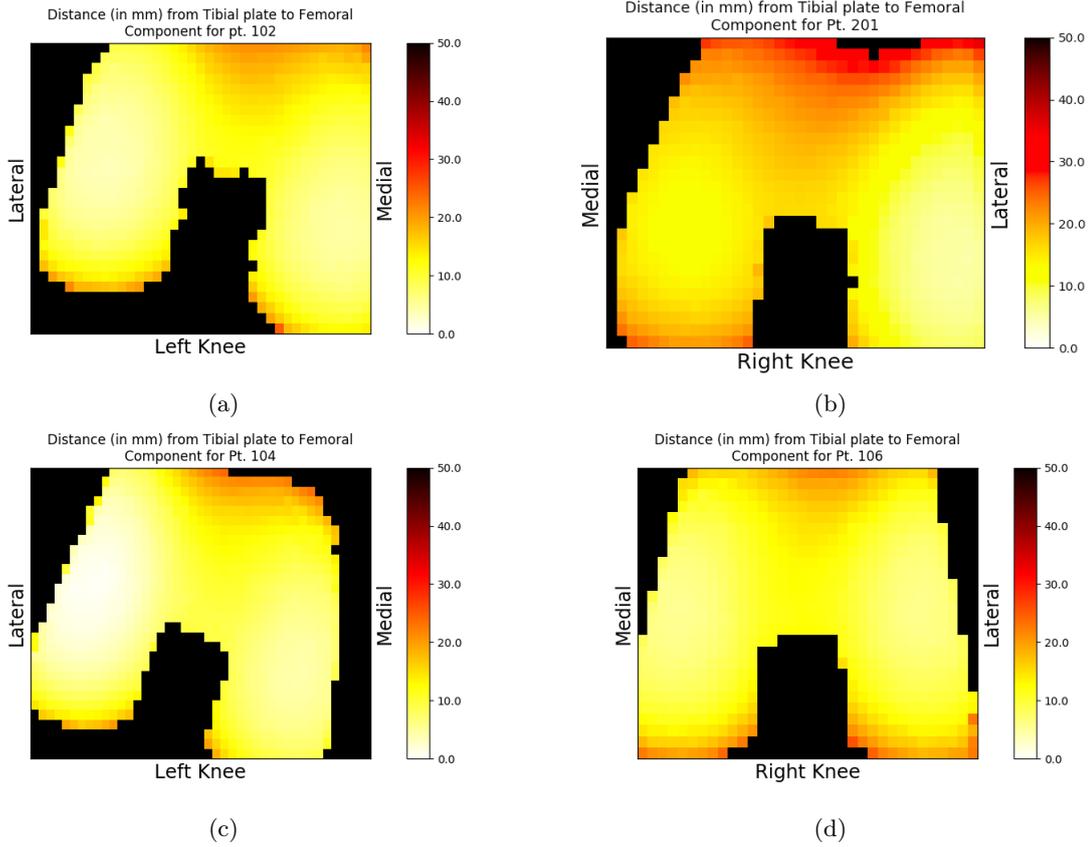


Figure 38: The above figures show examples of the liner wear measurements. In order, a) pt. 102, b) pt. 201, c) pt. 104, d) pt. 106.

Figures 39 show the liner wear results for data sets 107a and 107b. As is expected, the shape of the result of data set 107b is similar as for data set 107a, which is expected as it concerns the same patient and the position of the patient in the scanner has not changed, only the orientation of the scanner has changed, from upright to horizontal. The results are visually similar, except for the overall higher values in the result for data set 107b. This overall higher value is reflected in the mean and minimum values of the two data sets, while the standard deviations remain fairly similar. Pt. 107a: $14.28 \pm 4.42mm$, minimum $7.92mm$, pt. 107b: $18.71 \pm 5.02mm$, minimum $11.06mm$.

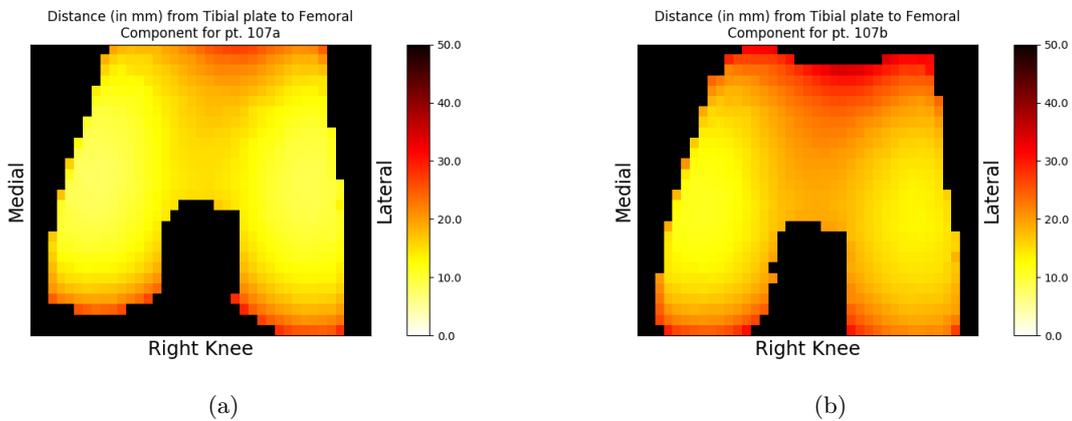


Figure 39: The figures above show the results for the vertical (patient 107a) a), and the supine (patient 107b) b) measurement.

7.3 Discussion

When viewing the results from the liner wear measurements, it is clear that the ideally aligned result (figure 37) has the same visual features, such as the "v-shape" and the hole between the condyles, as the other results. The major difference between the ideally aligned measurement and the rest of the data is that the ideally aligned measurement result is performed with the components directly above each other and ideally aligned in coronal, sagittal and transverse planes. Considering the slight flex of the knee of the patient when in the scanner, the components will most likely not be exactly aligned with each other accounting for part of the different appearances of the two measurements. Furthermore, translation and rotation of the femoral component is likely due to other pathological processes such as malalignment, liner wear or prosthetic migration. The difference is mostly positional, however, because the average values and standard deviations for the ideally aligned measurement and the other results are fairly similar, $11.31 \pm 5.18mm$ and $11.75 \pm 4.68mm$, respectively.

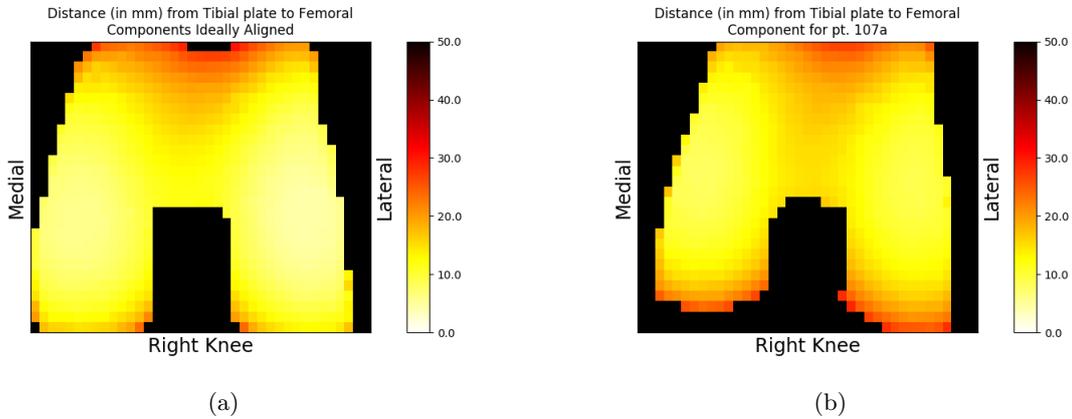


Figure 40: a)The ideally aligned result , and b) the measurement result for patient 107a .

It should be noted that the patients with the left-sided total knee prostheses both have lower mean and minimum values than the other patients, suggesting these patients have more wear than the right-sided patients. It is not clear what may cause this difference, as the prosthesis age, liner size and patient age do not differ from the other healthy controls. The increased liner wear may be due to increased activity of the patient, however patient activity was not determined, so this remains unclear. Furthermore, both are male patients, while only pt. 103 is also male in this patient data set. These are singular findings, however, and to determine whether this holds true more data needs to be analyzed.

Another observation can be made from the results of the measurement for data sets 107a and 107b. As can be seen from figure 39, the measurements for 107a and 107b show the same shape of the liner. The standard deviation of the data is also similar, however the average value of the data is $4.5mm$ higher for the results of 107b than for 107a. This can also be seen in the image, that the result of 107b has more points with a higher value than 107a. This suggests a difference between the vertical and supine measurements. This may play a role in clinical measurements, because in clinical practice MRI exams in supine position is the standard, while the clinical standard imaging is a standing long-leg radiograph. It may provide more insight in the comparisons between standard clinical x-ray imaging and supine imaging methods such as MRI and CT. The indicated difference between supine and upright scanning positions in this study should be investigated further and standardized to allow better comparison between upright and supine imaging results.

Currently, the results of the liner wear measurement consists of a heatmap of the distance between the femoral and tibial prosthesis components, as well as the mean, standard deviation and minimum values of the presented data. Ideally, the liner wear is measured over time and given as a value per unit of time, such as mm/y . when quantitative assessment is provided, comparison with the literature can be made to determine clinically significant liner wear [31, 43]. However, to accomplish the quantification of the liner wear over time, the measurement must first

be performed at two different times. Furthermore, to subtract the two data sets from one another, the position of the femoral and tibial components must be rotated to be exactly similar in both measurements, except in the direction perpendicular to the tibial plate grid. This is difficult to realize, as the components may change their position over time and therefore alter their alignment with respect to each other and the respective bones. To align the components with each other presents other difficulties when considering the spatial information obtained from the measurement should be maintained. A registration method reverts the possible change in the data over time, which makes it not suitable for to accomplish this task. Considering these difficulties, this last step of the liner wear method was not implemented considering the duration of this study.

8 General Discussion

Current clinical diagnosis of underlying pathologies of problematic primary total knee prosthesis requires many different imaging modalities. This study was part of an initiative to determine whether low-field MRI could provide a singular imaging alternative for the simultaneous diagnosis of problematic primary total knee prosthesis pathologies. From this study, it is seen that malalignment and liner wear can be determined from low-field MRI data, however, the large range of values in the malalignment and liner wear results compared to the clinically relevant reference data indicates the methods presented here are not yet suitable for clinical applications. Here, general discussion points of this study are presented.

A major strength of this study is that it provides a universal method for the measurement of malalignment and liner wear, using only scans of the relevant knee of the patient as well as the 3D models of the total knee prosthesis components implanted in the patient. The method is not concerned with whether the scans are CT or MRI, as long as the input data is of .stl file format. This universality of the method is ideal when considering the evolution of scanner types and the ongoing research on new types of total knee prostheses [102,103]. Provided that the input files for the methods are in .stl file format, the measurement will run.

The methods presented in this study all assume an ideal segmentation of the MR data. As can be seen from figure 15, many artifacts are still present which complicate and decrease the accuracy and reproducibility of the segmentation, see also table 4. An increase in image quality, with higher resolution and less artifacts, has multiple possible advantages [84]. When increased image quality is obtained, the borders of the prosthesis components become more defined which increases segmentation accuracy. Metal artifact reduction sequences should be considered to decrease the susceptibility artifacts of the prosthesis components in the images. Li *et al.* (2016) [48] have shown segmentation of the polyethylene liner in total knee arthroplasty is possible using standard clinical MRI scanners with a MAVRIC sequence. This type of sequence provides superior image quality [84,85,90] for more accurate image segmentation. Less pronounced susceptibility artifacts may provide enough improvement to effectively implement (semi-)automatic segmentation methods. These methods then increase the accuracy and reproducibility of the image segmentations [88,89] and further improve the measurement results.

The models used for these measurements are approximations created by scaling the scanned models to the measurements provided by [80]. These models are used for the registration to the segmentation data. However, the models used are not the same as the implanted prosthesis components. The models used are from Smith and Nephew, while the implanted components are from Zimmer-Biomet. Using the correct models for the registration will increase the registration accuracy of the data and will subsequently result in more accurate measurements.

The results from the malalignment measurements provide varying results with respect to the comparison of pathological total knee prosthesis and healthy controls. When comparing the results with the clinical reference values, the 95% CI of the study data shows some correspondence. However, some of the measurement results still fall outside of this range. The symptomatic knee patient *e.g.* has a cTA value outside the 95%CI range, while it does fall within the found ranges of the data used to calculate that interval. This suggests the data may not be of enough quantity to perform adequate statistical analysis. What the data from this study does suggest, however, is that the malalignment measurement results in many values at or close to the clinically relevant reference values for the data of the healthy controls. Considering the patients included in this study were not scanned directly post-operatively and therefore will most likely have a higher variance in measurement results due to migration and liner wear processes, the findings from this study suggest that the malalignment method implemented here is promising as a clinically relevant imaging method for the measurement of the malalignment pathology.

The clinical reference range for the sFA shows the least correspondence with the data from this study, however, when including the results from the literature [98], it can be seen that the malalignment measurement can be used to determine clinically relevant sFA of patients with a primary total knee prosthesis. This further supports the above statement that the malalignment measurement method may provide clinical benefits.

While a comparison is made to clinically relevant reference values for the malalignment measurement results, this comparison is premature. The clinically relevant reference values refer to ideal directly post-operative angles for the components [32,33,37,38], while our data of the healthy controls have an average prosthesis age of almost 1.5 years. During this time, pathological processes such as liner wear or migration may affect alignment of the prosthesis components. As such, the range of malalignment values found in this study is expected to be larger than when these patients would be scanned directly post-operatively. To allow more adequate comparison of the results with the values reported in literature and routine clinical imaging, directly post-operative MRI data should be obtained.

9 Recommendations

Viewing the points presented in the discussion above, several improvements should be made before clinical application can be considered.

To determine whether the current measurement provides clinically relevant values, a direct comparison with literature and standard clinical imaging should be obtained. Currently, the standard clinical imaging for malalignment consists of a standing long-leg x-ray [37]. To compare the clinical standard with the new method presented in this study, a data set of patients scanned directly post-operatively with both low-field MRI and x-ray should be obtained. This would determine the accuracy of the measurement and provide more adequate comparison with the clinical reference. Furthermore, this can also provide more insight in the value of low-field MRI as an alternative to current standard clinical imaging of problematic total knee prosthesis, which is currently unclear. To determine the current value of the measurements in this study, a standing long-leg radiograph should be made and malalignment and liner wear measurements should be performed on those images. This was not performed in this study, because the standing long-leg radiographs were not available for the patients included in this study. As such, a direct comparison could not be made.

When considering the low-field MRI data, it is seen that the image quality is lower than expected. The inaccuracies of the segmentation method provide a measurement accuracy that remains present in the registration and the malalignment and liner wear measurement. The influence of a bad segmentation can result in a . Therefore, a good segmentation of the imaging data is of crucial importance for the measurement. To increase the segmentation accuracy, the image quality should be increased by implementing a metal artifact reduction sequence such as MAVRIC or SEMAC-VAT [84,85,90]. Furthermore, segmentation reproducibility and possibly segmentation accuracy may be increased by implementing a (semi-)automatic segmentation method [89,104]. These methods can decrease the segmentation inaccuracies resulting from the current manual segmentation method combined with the low-field MRI results.

The results of the measurements for liner wear show that there is no difference between the results of the average healthy control compared to the patient with complaints of the total knee prosthesis. For the malalignment measurement, only the cFA was outside the range of the values calculated for the healthy controls. This may indicate that the patient with complaints has pathological total knee prosthesis based on another pathology than malalignment or liner wear. However, it may also indicate this method is not very applicable for the malalignment and liner wear. Considering the other values, the single pathological case, and the unknown origin of the knee complaints, however, rejecting the presented methods for the measurements of malalignment and liner wear does not seem obvious.

Furthermore, a comparison to clinically relevant reference values is not made for the liner wear measurement. This was not performed, because it was very difficult to realize the quantification of the liner wear. To quantify the extent of the liner wear, two measurements must be first be made. In the case of the absence of an earlier measurement, the ideally aligned measurement of the corresponding femur, tibia and liner size could be used. The measurement currently shows a heatmap of the distance between the femoral and tibial component. This provides spatial information that can provide clinicians with information about the location of possible liner wear. As such, the quantification should ideally retain the spatial information of the measurement. To perform the quantification, however, the two measurements should be exactly the same in shape and the position of the heatmap should also be the same. This allows the subtraction of the two measurements, but also presents the main difficulties of the quantification of the liner wear. Currently, the measurement only provides a calculation of the distance between the femoral and tibial prosthesis components. This can already provide clinicians with an intuition of the state of the liner. Nevertheless, before clinical implementation, the accuracy of the measurement should be determined with a comparison with clinical standard imaging and reference values.

10 Conclusion

From this work it can be concluded that, with the described methods, low-field MRI may provide an alternative method for the inspection of malalignment and liner wear in patients with total knee arthroplasty. However, comparison with clinical standards should be determined for both the malalignment and liner wear methods before clinical application. In this study, the results of the malalignment measurements were similar to the clinical reference ranges found in the literature, however, there was a larger range in the results of this study. Furthermore, the findings from this research show that a difference in the results from supine and standing scanning positions is present for both the malalignment and liner wear measurements. Also, image quality of low-field MRI data was lower than expected. Susceptibility artifacts present in the image made manual segmentation of the structures from the image data difficult and similarity measures show a less than good agreement between manual segmentations. Future studies should focus on the improvement of image quality, image segmentation, increasing the study population for a more clinically relevant comparison, and the difference between supine and standing low-field MRI exams.

References

- [1] J. Perry and B. Schoneberger, "GAIT ANALYSIS Normal and Pathological Function,"
- [2] S. N. Issa and L. Sharma, "Epidemiology of Osteoarthritis: An Update," *Current Rheumatology Reports*, vol. 8, pp. 7–15, 2006.
- [3] S. M. Hussain, D. W. Neilly, S. Baliga, S. Patil, and R. M. Meek, "Knee osteoarthritis: A review of management options," 2016.
- [4] N. Arden and M. C. Nevitt, "Osteoarthritis: Epidemiology," 2006.
- [5] S. J. Dixon, R. S. Hinman, M. W. Creaby, G. Kemp, and K. M. Crossley, "Knee joint stiffness during walking in knee osteoarthritis," *Arthritis Care & Research*, 2010.
- [6] "Richard D. Scott-Total Knee Arthroplasty -Saunders (2005),"
- [7] A. Chhabra, C. C. Elliott, and M. D. Miller, "Normal Anatomy and Biomechanics of the Knee,"
- [8] G. Li, S. Zayontz, E. Most, L. E. DeFrate, J. F. Suggs, and H. E. Rubash, "In situ forces of the anterior and posterior cruciate ligaments in high knee flexion: An in vitro investigation," *Journal of Orthopaedic Research*, 2004.
- [9] Fu F.H., C. D. Harner, and D. Johnson, "Biomechanics of knee ligaments.," *J Bone Joint Surg [Am]*, vol. 75, p. 17161727, 1993.
- [10] M. P. M. Steultjens, J. Dekker¹, M. E. Van Baar, R. A. B. Oostendorp², and J. W. J. Bijlsma³, "Range of joint motion and disability in patients with osteoarthritis of the knee or hip," *Rheumatology*, vol. 39, pp. 955–961, 2000.
- [11] C. W. Hutton, "Osteoarthritis: The cause not result of joint failure?," *Annals of the Rheumatic Diseases*, 1989.
- [12] "Online LROI annual report 2017 10 years of registration, a wealth of information,"
- [13] S. Kurtz, K. Ong, E. Lau, F. Mowat, and M. Halpern, "Projections of Primary and Revision Hip and Knee Arthroplasty in the United States from 2005 to 2030," *J Bone Joint Surg*, vol. 89-A, no. 4, pp. 780–785, 2007.
- [14] S. M. Kurtz, K. L. Ong, E. Lau, M. Widmer, M. Maravic, E. Gómez-Barrena, M. De Fátima De Pina, V. Manno, M. Torre, W. L. Walter, R. De Steiger, R. G. Geesink, M. Peltola, and C. Röder, "International survey of primary and revision total knee replacement," *International Orthopaedics*, 2011.
- [15] "Persona  The Personalized Knee."
- [16] J. C. Richmond, "Surgery for Osteoarthritis of the Knee," *Rheum Dis Clin N Am*, vol. 34, pp. 815–825, 2008.
- [17] D. E. Font-Rodriguez, G. R. Scuderi, and J. N. Insall, "Survivorship of Total Knee Arthroplasty," *Clinical Orthopaedics and Related Research*, vol. 345, pp. 79–86, 1997.
- [18] L. Guo, L. Yang, J. L. Briard, X.-j. Duan, and F.-y. Wang, "Long-term survival analysis of posterior cruciate-retaining total knee arthroplasty," *Knee Surgery, Sports Traumatology, Arthroscopy*, 2012.
- [19] M. B. Vessely, A. L. Whaley, W. S. Harmsen, C. D. Schleck, and D. J. Berry, "Long-term Survivorship and Failure Modes of 1000 Cemented Condylar Total Knee Arthroplasties Level of Evidence: Level IV, therapeutic study. See Guide- lines for Authors for a complete description of levels of evi- dence," *Clinical Orthopaedics and Related Research*, vol. 452, pp. 28–34, 2006.
- [20] D. J. Rodricks, S. Patil, P. Pulido, and C. W. Colwell, "Press-Fit Condylar Design Total Knee Arthroplasty Fourteen to Seventeen-Year Follow-up," *J Bone Joint Surg*, vol. 89-A, no. 1, pp. 89–95, 2007.

- [21] P. Rissanen, S. Aro, P. Slätis, H. Sintonen, and P. Paavolainen, “Health and quality of life before and after hip or knee arthroplasty,” *The Journal of Arthroplasty*, 1995.
- [22] M. Brandes, M. Ringling, C. Winter, A. Hillmann, and D. Rosenbaum, “Changes in physical activity and health-related quality of life during the first year after total knee arthroplasty,” *Arthritis Care and Research*, 2011.
- [23] Zimmer Biomet, “Surgical Technique for Legacy  Posterior Stabilized Knees NexGen.”
- [24] R. B. Bourne and P. G. Kirk, “Revision Total Knee Arthroplasty,”
- [25] R. B. Bourne and H. A. Crawford, “MANAGEMENT OF BONE LOSS DURING REVISION HIP OR KNEE REPLACEMENT PRINCIPLES OF REVISION TOTAL KNEE ARTHROPLASTY,”
- [26] A. Cheung, S. K. Goh, A. Tang, and T. B. Keng, “Complications of total knee arthroplasty,” *Current Orthopaedics*, 2008.
- [27] H. Mulcahy and F. S. Chew, “Current concepts in knee replacement: Complications,” 2014.
- [28] P. F. Sharkey, P. M. Lichstein, C. Shen, A. T. Tokarski, and J. Parvizi, “Why are total knee arthroplasties failing today-has anything changed after 10 years?,” *Journal of Arthroplasty*, 2013.
- [29] W. C. Schroer, K. R. Berend, A. V. Lombardi, C. L. Barnes, M. P. Bolognesi, M. E. Berend, M. A. Ritter, and R. M. Nunley, “Why are total knees failing today? Etiology of total knee revision in 2010 and 2011,” *Journal of Arthroplasty*, 2013.
- [30] D. F. Dalury, D. L. Pomeroy, R. S. Gorab, and M. J. Adams, “Why are total knee arthroplasties being revised?,” *Journal of Arthroplasty*, 2013.
- [31] A. Hoshino, Y. Fukuoka, and A. Ishida, “Accurate in vivo measurement of polyethylene wear in total knee arthroplasty,” *The Journal of Arthroplasty*, vol. 17, no. 4, pp. 490–496, 2002.
- [32] J. J. Cherian, B. H. Kapadia, S. Banerjee, J. J. Jauregui, K. Issa, and M. A. Mont, “Mechanical, anatomical, and kinematic axis in TKA: Concepts and practical applications,” in *Current Reviews in Musculoskeletal Medicine*, 2014.
- [33] M. P. Abdel, S. Oussedik, S. Parratte, S. Lustig, and F. S. Haddad, “Coronal alignment in total knee replacement: historical review, contemporary analysis, and future direction,” *The Bone & Joint Journal*, 2014.
- [34] M. Schiraldi, G. Bonzanini, D. Chirillo, and V. de Tullio, “Mechanical and kinematic alignment in total knee arthroplasty,” *Annals of Translational Medicine*, 2016.
- [35] T. L. Petersen and G. A. Engh, “Radiographic assessment of knee alignment after total knee arthroplasty,” *Journal of Arthroplasty*, 1988.
- [36] E. L. Feng, S. D. Stulberg, and R. L. Wixson, “Progressive subluxation and polyethylene wear in total knee replacements with flat articular surfaces,” *Clinical Orthopaedics and Related Research*, vol. 299, pp. 60–71, 1994.
- [37] F. C. Ewald, “The Knee Society Total Knee Arthroplasty Roentgenographic Evaluation and Scoring System,”
- [38] C. Y. Lee, S. J. Lin, L. T. Kuo, K. T. Peng, K. C. Huang, T. W. Huang, M. S. Lee, R. W. W. Hsu, and W. J. Shen, “The benefits of computer-assisted total knee arthroplasty on coronal alignment with marked femoral bowing in Asian patients,” *Journal of orthopaedic surgery and research*, 2014.
- [39] G. Stan, H. Orban, L. Gruionu, and P. Gheorghe, “Coronal malposition effects in total knee arthroplasty: a finite element analysis,” *European Journal of Orthopaedic Surgery & Traumatology*, 2013.

- [40] F. H. Sim, J. H. Beaty, S. T. Canale, D. C. Ferlic, D. L. Helfet, M. J. Archibeck, J. J. Jacobs, K. A. Roebuck, and T. T. Glant, “The Basic Science of Periprosthetic Osteolysis,” *J Bone Joint Surg*, vol. 82-A, no. 10, pp. 1478–1489, 2000.
- [41] S. B. Goodman, E. Gibon, and Z. Yao, “The Basic Science of Periprosthetic Osteolysis,” *Instr Course Lect*, vol. 62, pp. 201–206, 2013.
- [42] Y. Zhu, K. Chiu, and W. Tang, “Review Article: Polyethylene wear and osteolysis in total hip arthroplasty,” *Journal of Orthopaedic Surgery*, 2001.
- [43] J. Benjamin, J. Szivek, G. Dersam, S. Persselin, and R. Johnson, “Linear and Volumetric Wear of Tibial Inserts in Posterior Cruciate-Retaining Knee Arthroplasties,” *Clinical Orthopaedics and Related Research*, vol. 392, pp. 131–138, 2001.
- [44] J. P. Collier, M. B. Mayor, J. L. McNamara, V. A. Surprenant, and R. E. Jensen, “Analysis of the failure of 122 tibial inserts from uncemented tibial knee components,” *Clinical Orthopaedics and Related Research*, vol. 273, pp. 232–242, 1991.
- [45] G. Chakrabarty, M. Vashishtha, and D. Leeder, “Polyethylene in knee arthroplasty: A review,” 2015.
- [46] C. M. Sofka, R. S. Adler, and R. Laskin, “Sonography of polyethylene liners used in total knee arthroplasty,” *American Journal of Roentgenology*, 2003.
- [47] R. Guillin, J.-L. Laporte, P. Sabouret, E. Cardinal, and P. E. Cardinal, “Polyethylene Wear in Knee Arthroplasty A New Sonographic Sign,” *Musculoskeletal Imaging*, 2007.
- [48] A. E. Li, D. B. Sneag, T. T. Miller, J. D. Lipman, D. E. Padgett, and H. G. Potter, “MRI of polyethylene tibial inserts in total knee arthroplasty: Normal and abnormal appearances,” *American Journal of Roentgenology*, 2016.
- [49] Y. Abu-Amer, I. Darwech, and J. C. Clohisy, “Aseptic loosening of total joint replacements: mechanisms underlying osteolysis and potential therapies,” *Arthritis Research & Therapy*, 2007.
- [50] H.-N. Pang, H. R. B. Abd Razak, S. Petis, D. D. R. Naudie, and S. J. MacDonald, “The role of isolated polyethylene exchange in total knee arthroplasty,” *EFORT Open Reviews*, 2017.
- [51] W. Petersen, I. V. Rembitzki, G.-P. Brüggemann, A. Ellermann, R. Best, A. G. Koppenburg, and C. Liebau, “Anterior knee pain after total knee arthroplasty: a narrative review,” *International Orthopaedics*, 2014.
- [52] Paolo Russo, ed., *Handbook of X-ray Imaging: Physics and Technology*. Napels: CRC Press Taylor & Francis Group, 2018.
- [53] R. Fosbinder and D. Orth, “Computed Tomography,” in *Essentials of Radiologic Science*, ch. 17, p. 263, Lippincott Williams & Wilkins, 2011.
- [54] L. Ryd, “Roentgen Stereophotogrammetric Analysis of Prosthetic Fixation of the Hip and Knee Joint,” *Clinical Orthopaedics and Related Research*, no. 276, pp. 56–65, 1992.
- [55] L. Ryd, B. E. J. Albrektsson, L. Carlsson, F. Dansgard, P. Herberts, A. Lindstrand, L. Regner, and S. Toksvig-Larsen, “Roentgen Stereophotogrammetric Analysis as a Predictor of Mechanical Loosening of Knee Prostheses,” *J Bone Joint Surg*, vol. 77, no. 3, pp. 377–83, 1995.
- [56] J. Kärrholm, R. H. S Gill, and E. R. Valstar, “The History and Future of Radiostereometric Analysis,” *Clinical Orthopaedics and Related Research*, vol. 448, pp. 10–21, 2006.
- [57] E. R. Valstar, R. G. H. H. Nelissen, J. H. C. Reiber, and P. M. Rozing, “The use of Roentgen stereophotogrammetry to study micromotion of orthopaedic implants,” *ISPRS Journal of Photogrammetry and Remote Sensing*, 2002.
- [58] G. Selvik, “Roentgen stereophotogrammetry,” *Acta Orthopaedica Scandinavica*, 1989.

- [59] P. Maravi, D. Maravi, S. Uikey, and L. Kaushal, “Role of MRI in Orthopedics,” *Orthopaedic Journal of M.P. Chapter*, vol. 21, no. 2, pp. 74–77, 2015.
- [60] G. Brix, *Magnetic Resonance Tomography*. Springer, 2008.
- [61] P. Sprawls, “Tissue Magnetization And Relaxation,” in *Magnetic Resonance Imaging*, ch. 4.
- [62] K. M. Koch, B. A. Hargreaves, K. B. Pauly, W. Chen, G. E. Gold, and K. F. King, “Magnetic Resonance Imaging Near Metal Implants,” *J. Magn. Reson. Imaging*, vol. 32, pp. 773–787, 2010.
- [63] A. D. Elster, “Sellar susceptibility artifacts: Theory and implications,” *American Journal of Neuroradiology*, 1993.
- [64] A. D. Elster, “Questions and Answers in MRI,” 2018.
- [65] “Spin Echo,” 2017.
- [66] J. F. Schenck, “The role of susceptibility in magnetic resonance imaging: MRI magnetic compatibility of the first and second kinds,” *Medical Physics*, vol. 23, no. 6, pp. 815–850, 1996.
- [67] S. Ghazinoor, J. V. Crues, and C. Crowley, “Low-field musculoskeletal MRI,” *Journal of Magnetic Resonance Imaging*, vol. 25, no. 2, pp. 234–244, 2007.
- [68] C. S. Lee, S. M. Davis, C. McGroder, S. Kouk, R. M. Sung, W. B. Stetson, and S. E. Powell, “Analysis of Low-Field MRI Scanners for Evaluation of Shoulder Pathology Based on Arthroscopy,” *Orthopaedic journal of sports medicine*, vol. 2, no. 7, p. DOI: 10.1177/2325967114540407, 2014.
- [69] K. Farahani, U. Sinha, S. Sinha, L. C-L Chiu, and R. B. Lufkin, “Effect of Field Strength on Susceptibility Artifacts in Magnetic Resonance Imaging,” *Computerized Medical Imaging and Graphics*, vol. 14, no. 6, pp. 409–13, 1990.
- [70] Z. Munn and Z. Jordan, “Interventions to reduce anxiety, distress and the need for sedation in adult patients undergoing magnetic resonance imaging: A systematic review,” *International Journal of Evidence-Based Healthcare*, 2013.
- [71] Z. Munn and Z. Jordan, “The patient experience of high technology medical imaging: A systematic review of the qualitative evidence,” *Radiography*, 2011.
- [72] V. Tischler, T. Calton, M. Williams, and A. Cheetham, “Patient anxiety in magnetic resonance imaging centres: Is further intervention needed?,” *Radiography*, 2008.
- [73] A. Abragam, *The Principles of Nuclear Magnetism*. 1961.
- [74] R. K. L. Lee, J. F. Griffith, Y. Y. O. Lau, J. H. Y. Leung, A. W. H. Ng, E. H. Y. Hung, and S. W. Law, “Diagnostic Capability of Low- Versus High-Field Magnetic Resonance Imaging for Lumbar Degenerative Disease,” *Spine*, vol. 40, no. 6, pp. 382–391, 2015.
- [75] A. M. Coffey, M. L. Truong, and E. Y. Chekmenev, “Low-field MRI can be more sensitive than high-field MRI,” *Journal of Magnetic Resonance*, vol. 237, pp. 169–174, 2013.
- [76] A. Nagy and S. Dyson, “Magnetic resonance anatomy of the carpus of the horse described from images acquired from low-field and high-field magnets,” *Veterinary Radiology and Ultrasound*, vol. 52, no. 3, pp. 273–283, 2011.
- [77] R. C. Murray, T. S. Mair, C. E. Sherlock, A. S. Blunden R C Murray, and A. S. Blunden, “Comparison of high-field and low-field magnetic resonance images of cadaver limbs of horses,” *Veterinary Record*, vol. 165, pp. 281–288, 2009.
- [78] B. Kladny, K. Glückert, B. Swoboda, W. Beyer, and G. Weseloh, “Comparison of low-field (0.2 Tesla) and high-field (1.5 Tesla) magnetic resonance imaging of the knee joint,” *Archives of Orthopaedic and Trauma Surgery*, 1995.

- [79] “G-Scan Brio: The Key to Confidence,” tech. rep.
- [80] Genesis, “Overview of the LEGION Total Knee System Seamless system, multiple options LEGION HK Hinge Knees Identical trochlear groove geometry preserved throughout all LEGION knees Consistent femoral A/P box resections (HK = RK with 10mm posterior wedge resections),” tech. rep.
- [81] T. Migimatsu, “Automatic MRI Bone Segmentation,”
- [82] V. Pedoia, S. Majumdar, and T. M. Link, “Segmentation of joint and musculoskeletal tissue in the study of arthritis,” 2016.
- [83] C. A. Chen, W. Chen, S. B. Goodman, B. A. Hargreaves, K. M. Koch, W. Lu, A. C. Brau, C. E. Draper, S. L. Delp, and G. E. Gold, “New MR imaging methods for metallic implants in the knee: Artifact correction and clinical impact,” *Journal of Magnetic Resonance Imaging*, 2011.
- [84] K. M. Koch, A. C. Brau, W. Chen, G. E. Gold, B. A. Hargreaves, M. Koff, G. C. McKinnon, H. G. Potter, and K. F. King, “Imaging near metal with a MAVRIC-SEMAC hybrid,” *Magnetic Resonance in Medicine*, 2011.
- [85] T. Ai, A. Padua, F. Goerner, M. Nittka, Z. Gugala, S. Jadhav, M. Trelles, R. F. Johnson, R. W. Lindsey, X. Li, and V. M. Runge, “SEMAC-VAT and MSVAT-SPACE Sequence Strategies for Metal Artifact Reduction in 1.5T Magnetic Resonance Imaging,” *Investigative Radiology*, 2012.
- [86] “Design Rationale Vanguard $\text{\textcircled{R}}$ Complete Knee System,” tech. rep.
- [87] J. J. Bartko, “Measurement and reliability: Statistical thinking considerations,” *Schizophrenia Bulletin*, 1991.
- [88] T. Stammberger, F. Eckstein, M. Michaelis, K. H. Englmeier, and M. Reiser, “Interobserver reproducibility of quantitative cartilage measurements: Comparison of B-spline snakes and manual segmentation,” *Magnetic Resonance Imaging*, 1999.
- [89] A. Bazille, M. A. Guttman, E. McVeigh, and E. A. Zerhouni, “Impact of Semiautomated versus Manual Image Segmentation Errors on Myocardial Strain Calculation by Magnetic Resonance Tagging,” *Investigative Radiology*, vol. 29, no. 4, pp. 427–433, 1994.
- [90] C. L. Hayter, M. F. Koff, P. Shah, K. M. Koch, T. T. Miller, and H. G. Potter, “MRI after arthroplasty: Comparison of MAVRIC and conventional fast spin-echo techniques,” *American Journal of Roentgenology*, 2011.
- [91] H. P. A. Künzi and B. Sendov, “Hausdorff distance and image processing,” *Russ. Math. Surv.*, vol. 59, p. 319, 2004.
- [92] D. L. Miranda, M. J. Rainbow, E. L. Leventhal, J. J. Crisco, and B. C. Fleming, “Automatic determination of anatomical coordinate systems for three-dimensional bone models of the isolated human knee,” *Journal of Biomechanics*, 2010.
- [93] T. Walker, “Convex Hull.”
- [94] J. Stanton, “Computing Areas,” 2014.
- [95] P. Bourke, “Polygons and meshes,” 1988.
- [96] Gustavb, “Unit Circle,” 2006.
- [97] D. M. Fang, M. A. Ritter, and K. E. Davis, “Coronal Alignment in Total Knee Arthroplasty. Just How Important is it?,” *Journal of Arthroplasty*, 2009.
- [98] V. Kilincoglu, K. Unay, K. Akan, I. Esenkaya, and O. Poyanli, “Component alignment in simultaneous bilateral or unilateral total knee arthroplasty,” *International Orthopaedics*, 2011.

- [99] M. Hadi, T. Barlow, I. Ahmed, M. Dunbar, P. McCulloch, and D. Griffin, “Does malalignment affect revision rate in total knee replacements: a systematic review of the literature,” *SpringerPlus*, 2015.
- [100] “Surgical Technique NexGen Prime Knee MIS Multi-Reference [®] 4-in-1 Femoral Instrumentation For NexGen [®] Cruciate Retaining & NexGen Legacy [®] Posterior Stabilized Knees.”
- [101] H. Pao, R. Gallotta, and A. Daigavane, “Distance between Point and Line.”
- [102] J.-L. Prudhon and R. Verdier, “Cemented or cementless total knee arthroplasty?,” *SICOT-J*, 2017.
- [103] D. F. Dalury, “Cementless total knee arthroplasty,” *Bone and Joint Journal*, 2016.
- [104] T. Stammberger, F. Eckstein, M. Michaelis, K.-H. Englmeier, and M. Reiser, “Interobserver reproducibility of quantitative cartilage measurements: comparison of B-spline snakes and manual segmentation,” *Magnetic Resonance Imaging*, vol. 17, no. 7, pp. 1033–1042, 1999.

11 Appendix A: Liner Wear Python Code

```
"""This code determines the distance between the femoral and tibial
prosthesis components. To run this code, several things are required.
First, the segmentations of the femur and tibia bones. In this code,
the segmentations are expected to be in .stl format. Also,
the components are expected to be in .stl format. The registration
matrix that is referred to in this code is the registration matrix that
results from the registration in Mimics and should be in .txt format.
```

```
The loading of the data is set up to accommodate the current method of
saving data in the file and data structure employed in the accompanying
data folder. When changing this, the input values and data loading paths
should be changed accordingly."""
```

```
"""IMPORT THE REQUIRED PACKAGES"""
```

```
import time
start_time = time.time()
import numpy as np
from stl import mesh #stl = numpystl package
from scipy.spatial import ConvexHull
```

```
"""DEFINE THE REQUIRED FUNCTIONS"""
```

```
def ImportData(path):
```

```
    """The function ImportData imports the mesh file and the center of
    gravity of the mesh from the given path to the .stl file.
```

```
    Input: path: path to the .stl file of the measurement data.
```

```
    Output: mesh: the mesh determine from the file path.
```

```
           center of gravity: the center of gravity of the mesh file.
```

```
    """
```

```
    meshFile = mesh.Mesh.from_file(path)
    [volume, center_of_gravity, inertia] = \
        mesh.Mesh.get_mass_properties(meshFile)
```

```
    return meshFile
```

```
def LinearInterp(xmax, xmin, ymax, ymin, gridPoint_distance):
```

```
    """Function LinearInterp, determine the linear interpolation of the
    other component of the x and y data of the largest components of the
    convex hull.
```

```
    Input:  xmax - maximum x value for the grid
```

```
           xmin - minimum x value for the grid
```

```
           ymax - maximum y value for the grid
```

```
           ymin - minimum y value for the grid
```

```
           gridpoint_distance - step distance between each point.
```

```
    Output: X - x values of the gridpoints
```

```
           Y - y values of the gridpoints
```

```
           xComps - vector with the values along the x-direction
```

```
           yComps - vector with the values along the y-direction"""
```

```

xComps = np.arange(xmin-1, xmax+1, gridPoint_distance)
yComps = np.arange(ymin-1, ymax+1, gridPoint_distance)

X = []
Y = []

for x in xComps:
    for y in yComps:
        X.append(x)
        Y.append(y)

return X, Y, xComps, yComps

def findMaxima(mesh):
    """The function findMaxima determines the maximum and minimum x and y
    values in 3D space. This is done by determining the maxima and
    minima of the input mesh.
    Input: mesh - mesh of which to determine the maxima and minima
    Output: maxVertx - maximum x coordinate
           minVertx - minimum x coordinate
           maxVertY - maximum y coordinate
           minVertY - minimum y coordinate"""

    minVertx = []
    minVertY = []

    maxVertx = []
    maxVertY = []

    normals = []

    for vertex in np.arange(mesh.vectors.__len__()):
        minVertexX = np.amin(mesh.vectors[vertex], axis=0)[0]
        minVertexY = np.amin(mesh.vectors[vertex], axis=0)[1]
        maxVertexX = np.amax(mesh.vectors[vertex], axis=0)[0]
        maxVertexY = np.amax(mesh.vectors[vertex], axis=0)[1]
        if minVertx == []:
            minVertx = minVertexX
        elif minVertexX < minVertx:
            minVertx = minVertexX
        if minVertY == []:
            minVertY = minVertexY
        elif minVertexY < minVertY:
            minVertY = minVertexY
        if maxVertx == []:
            maxVertx = maxVertexX
        elif maxVertexX > maxVertx:
            maxVertx = maxVertexX
        if maxVertY == []:
            maxVertY = maxVertexY
        elif maxVertexY > maxVertY:
            maxVertY = maxVertexY

    return maxVertx, minVertx, maxVertY, minVertY

def MaxHullGrid(mesh1, amount_bins, gridPoint_distance):
    """The function MaxHullGrid determines the position of a grid G and

```

its points. This is done by determining the convex hull with the most points in the pre-registration tibial mesh and setting the z-value of each point on G equal to the z-value of the calculated position of the convex hull.

Input: mesh1 - the tibial mesh from which the position of the grid is determined
 mesh2 - the femoral mesh, used for sizing of the grid
 amount_bins - the step-size along the Z-direction. The larger amount_bins, the more points are taken into account for determining the hull, hull area and hull centroid.
 gridPoint_distance - the distance between the points on the grid.

Output: gridX -
 gridY -
 gridSize -
 Cent_z -

"""

Get all the z-values of the mesh and determine the unique values,
 # the maximum and minimum values.

Z = mesh1.z
 Z_new = np.unique(Z)

Z_min = Z_new.min()
 Z_max = Z_new.max()

Set up the bins for future use
 start_bin = np.floor(Z_min)
 end_bin = np.ceil(Z_max)
 numbers = np.arange(start_bin, end_bin, amount_bins)

diff_bin = end_bin - start_bin

Make sure that all values of Z are taken into account
 if diff_bin % amount_bins != 0.0:
 last_num = end_bin + (amount_bins - diff_bin % amount_bins)
 numbers = np.append(numbers, last_num)

Set z-coordinate of centroid in the middle of the slice
 Cent_z = []
 for nums in np.arange(numbers.shape[0] - 1):
 Cent_z.append((numbers[nums] + numbers[nums + 1]) / 2)

mesh_vectors = mesh1.vectors

###-----###

This part of the loop determines the vertices that correspond to
 # the slices that will be used for the measurement as determined above.

Z_bins = []
 zbinSize = []
 # loop through bins
 for num in np.arange(numbers.shape[0] - 1):
 Z_bin = []
 # Loop through the numbers in the bins

```

for Z_num in Z_new:
    # If number inside bin range put in Z_bin
    if Z_num >= numbers[num]:
        if Z_num <= numbers[num + 1]:
            Z_bin.append(Z_num)
        zbinSize.append(Z_bin.__len__())
    Z_bins.append(Z_bin)

# Select largest bin (with the most vertices, aka the surface of the
# tibial component)
zbinSize = 0
indNr = 0
for bin in Z_bins:
    indNr += 1
    if not bin:
        continue
    elif bin.__len__() > zbinSize:
        zbinSize = bin.__len__()
        index = indNr

largestBin = Z_bins[index]

# Collect all the corresponding vectors in largestBin.
Z_temp = np.empty([1, 3])
for z_num in largestBin:
    for vector in mesh_vectors:
        if z_num in vector:
            Z_temp = np.concatenate([Z_temp, vector], 0)

###-----###

# Determine the Convex Hull of the set of points in largestBin
XY_temp = Z_temp[1:, 0:2]
hull = ConvexHull(XY_temp)
x = XY_temp[hull.vertices, 0]
y = XY_temp[hull.vertices, 1]

verts = XY_temp[hull.vertices]

# Calculate the maxima and minima for the Grid.
Xmins = verts[np.argmin(x)]
Xmaxs = verts[np.argmax(x)]
Ymins = verts[np.argmin(y)]
Ymaxs = verts[np.argmax(y)]
# sizes = PolyArea(x, y)

sizeX = [Xmins[0], Xmaxs[0]]
sizeY = [Ymins[1], Ymaxs[1]]

gridSize = [sizeX,
            sizeY]

xMax, xMin, yMax, yMin = findMaxima(mesh1)

gridX, gridY, xSize, ySize = LinearInterp(xMax, xMin, yMax, yMin,
                                           gridPoint_distance)

```

```

    return gridX, gridY, gridSize, Cent_z[index], xSize, ySize

def readTransform(path):
    """The function readTransform computes the rotation matrix form the
    input .txt file containing the 4x4 transformation matrix
    corresponding to the registration of the corresponding CAD model.

    Input: path: the path to the .txt file containing the transformation
    matrix.
    Output: new Z-axis vector.
    """

    file = open(path, 'r')
    data = file.readlines()

    def findNR(string):
        l = []
        tokens = string.split()
        for token in tokens:
            l.append(float(token))

        return l

    M1 = findNR(data[2])
    M2 = findNR(data[3])
    M3 = findNR(data[4])
    M4 = findNR(data[5])

    MM = [M1,
           M2,
           M3, ]

    TR = [M1,
           M2,
           M3,
           M4]

    T = [MM[0][3],
          MM[1][3],
          MM[2][3], ]

    inverseT = (-1)*T
    # print(inverseT)

    Sx = np.linalg.norm(MM[0])
    Sy = np.linalg.norm(MM[1])
    Sz = np.linalg.norm(MM[2])

    RotMat = [[MM[0][0] / Sx, MM[0][1] / Sx, MM[0][2] / Sx],
               [MM[1][0] / Sy, MM[1][1] / Sy, MM[1][2] / Sy],
               [MM[2][0] / Sz, MM[2][1] / Sz, MM[2][2] / Sz]]

    tranform_matrix = [[MM[0][0] / Sx, MM[0][1] / Sx, MM[0][2] /
                          Sx, MM[0][3]],
                        [MM[1][0] / Sy, MM[1][1] / Sy, MM[1][2] /
                          Sy, MM[1][3]],
                        [MM[2][0] / Sz, MM[2][1] / Sz, MM[2][2] /
                          Sz, MM[2][3]]]

```

```

GCS = [[1, 0, 0],
        [0, 1, 0],
        [0, 0, 1]]

LCS = np.matmul(GCS, RotMat)

New_x = LCS[0]
New_y = LCS[1]
New_z = LCS[2]

# transposeRotmat = np.transpose(RotMat)
inverseRotmat = np.linalg.inv(RotMat)
# print(inverseRotmat)
# print(np.matmul(transposeRotmat, RotMat))
# print(np.array(RotMat))
# TR = [[MM[0][0] / Sx, MM[0][1] / Sx, MM[0][2] / Sx, MM[0][3]],
#       [MM[1][0] / Sy, MM[1][1] / Sy, MM[1][2] / Sy, MM[1][3]],
#       [MM[2][0] / Sz, MM[2][1] / Sz, MM[2][2] / Sz, MM[2][3]],
#       [0, 0, 0, 1]]

# print(MM)

return [New_z], RotMat, T, MM, inverseRotmat, inverseT, np.array(TR)

def lineLength(lineP1, lineP2):
    """The function lineLength determines the length of the line between
    two point P1 and P2.
    Input: lineP1 - the starting point from which to determine the length
           of the line.
           lineP2 - the end point
    Output: length - the length of the line between lineP1 and lineP2"""

    x = lineP2[0][0] - lineP1[0][0]
    y = lineP2[1][0] - lineP1[1][0]
    z = lineP2[2][0] - lineP1[2][0]

    length = np.sqrt(np.square(x) + np.square(y) + np.square(z))

    return length

def rayplaneIntersect2(plane, linePoint0, linePoint1, normal):
    """The function rayplaneIntersect2 calculates the intersection of a
    ray with a plane.
    Input: plane - a 3x3 matrix with the coordinates of the points which
           determine the plane
           linePoint0 - Starting point on the grid
           linePoint1 - Second point, a point on the ray, determining the
           direction of the ray
           normal - the normal of the plane."""

    # Determine direction of the ray
    direction = linePoint1 - linePoint0

    planePoint1 = np.array([plane[0],
                            plane[1],

```

```

        plane[2]))

# planeNormal_unit = normal/np.linalg.norm(normal)
# planeX = [plane[0], plane[3], plane[6]]
# planeY = [plane[1], plane[4], plane[7]]
# planeZ = [plane[2], plane[5], plane[8]]

p1 = planePoint1

knum = np.dot(normal, (p1 - linePoint0).T)
kdenom = np.dot(normal, direction.T)

if kdenom == 0:
    intersectPoint = []
else:
    k = knum / kdenom
    intersectPoint = (linePoint0 + (k * direction))

return intersectPoint

def reduceHalf(mesh):
    """The function reduceHalf reduces input mesh by half by removing the
    upper half of the data.
    Input: mesh - the input mesh to be reduced
    Output: M2 - the reduced mesh
    -----"""

    z_del = []
    M = mesh
    # M2 = M
    cutoff = np.amin(M.z) + ((np.amax(M.z) - np.amin(M.z)) / 2)

    for Z in np.arange(M.z._len_()):
        for j in M.z[Z]:
            if j > cutoff:
                z_del.append(Z)
                continue

    M2 = np.delete(M, np.unique(z_del), 0)
    return M2

,-----,

"""INITIALIZE REQUIRED VARIABLES"""
,-----,

# SETS FOR PATH INITIALIZATION
femurSizeSet = ["AB", "CD", "EF", "GH"]
tibiaSizeSet = ["12", "34", "56", "7890"]

# EMPTY DATASETS FOR THE TRANSFORMED POINT DATA
XpointsStart = []
YpointsStart = []
ZpointsStart = []

```

```

XpointsEnd = []
YpointsEnd = []
ZpointsEnd = []

# EMPTY DATASETS FOR THE INTERSECTPOINT DATA
xInter = []
yInter = []
zInter = []

# EMPTY DATASETS FOR THE LENGTH DATA
analysisData = []
pointsData = []

# EMPTY DATASET FOR THE NORMALS OF THE DATA
normals = []

# EMPTY DATASET TO BE USED TO PERFORM CALCULATIONS ON THE LENGTH DATA
goodPoints = []

,_____

""SET REQUIRED INPUT VALUES""
,_____

""Give the required inputs""
#ptNumber the number of the dataset
ptNumber = "7"

if ptNumber.__len__(<2:
    ptNumber = "0"+ptNumber

#The left or right knee
side = "Right"

#Size of the femoral prosthesis component
FemurSize = "E"

#Size of the tibial prosthesis component
TibiaSize = "6"

#Size of the liner (the size on the box, not the articulating surface size)
LinerSize = "10"

# DISTANCE BETWEEN THE POINTS IN THE GRID
gridPoint_distance = 1

"For simpler data organization, I put the tibial and femoral components " \
"in folders that correspond to the sizes that can be provided for each " \
"liner size. Furthermore they are divided in left and right knees. The " \
"current setup of folders is: Left or Right\((Femur Size)\_(Tibia Size)\_( " \
"Liner Size)" \
"Each folder contains the femoral and tibial component models for the " \
"corresponding component sizes. The models are aligned such that the " \
"thickness of the liner (liner size) is between them. Thus, the distance" \
"between the component models in the folders corresponds to the size of " \

```

```

"liner given by the respective folder."

if TibiaSize == "10":
    TibiaSetSize = "0"
else:
    TibiaSetSize = TibiaSize

for set in femurSizeSet:
    for letter in set:
        if letter == FemurSize:
            femsetSize = set

for set in tibiaSizeSet:
    for number in set:
        if number == TibiaSetSize:
            tibsetSize = set

femSide = side[0]
FemurSideSize = FemurSize + femSide

```

```

""""SET DATA FOLDERS FOR EASY ACCESS""""

```

```

# SET HOME FOLDER

# !!! THIS FOLDER SHOULD BE CHANGED WHEN A NEW/DIFFERENT FOLDER LOCATION
# IS USED !!!

folderLocation = "/Volumes/L-BOX/Afstuderen"

# SET MAIN FOLDER AND SEGMENTATION AND REGISTRATION FOLDERS
mainFolder = folderLocation + "/Data"
subFolder_Segmentation = mainFolder + "/Segmentations"
subFolder_Registration = mainFolder + "/Registrations"

# Segmentation data is in segFolder. Except, pt.1 has 3 segmentations for
# ICC-calculation purposes. Therefore,
# segFolder = "Segmentation 1" instead of just "Segmentation".
# (The choice for using the 1st segmentation is ambiguous)

if ptNumber == 1:
    segFolder = "/Segmentation 1"
    regFolder2 = "/Registration 1"
else:
    segFolder = ""
    regFolder2 = ""

# Set ptFolder to extract patient specific image data.
ptFolder_seg = subFolder_Segmentation + "/Pt. 1{}".format(ptNumber,
                                                         segFolder)
ptFolder_reg = subFolder_Registration + "/Pt. 1{}".format(ptNumber,
                                                         regFolder2)

```

```

# Set path to corresponding model data.
modelFolder = mainFolder + "/Model Data/{}/F-{} T-{} L{}".format(side ,
                                                                    femsetSize , tibsetSize , LinerSize)
,

""""LOAD DATA FROM PATH STRINGS""""
,

""""The following section is to extract the correct data from the
corresponding folders. Unless other folder and file names are used,
nothing should be changed here.""""

# TIBIAL DATA
# Load tibial segmentations, registrations, and required 3D models

segTibpath = ptFolder_seg+"/Tibia/tibBone.Segmentation.stl".\
    format(ptNumber)
segTib = mesh.Mesh.from_file(segTibpath)

# Load tibial component segmentation
segTibcompPath = ptFolder_seg+"/Tibia/tibComp.Segmentation.stl".\
    format(ptNumber)
segTibcomp = mesh.Mesh.from_file(segTibcompPath)

# Path to unregistered tibial component
tib_Testpath = modelFolder+"/tibComp_zalign_size-{}.stl".\
    format(TibiaSize)
tib_mesh = mesh.Mesh.from_file(tib_Testpath)

# Load registered tibial component for the patient
registered_tibComp = ptFolder_reg+"/Tibia/tibComp_Registered.stl".\
    format(ptNumber)
reg_tibMesh = mesh.Mesh.from_file(registered_tibComp)

# Path to tibial registration matrix from Mimics
path_to_tibial_transform = ptFolder_reg+"/Tibia/Tibia Registration " \
    "Matrix.txt".format(ptNumber)
tib_comp_z , tibRotmat , tibTrans , tib_transform , inverse_tibRotmat , \
inverse_tibTrans , TR4 = readTransform(path_to_tibial_transform)

# FEMORAL DATA
# Load femoral segmentations, registrations and required 3D models.

segFempath = ptFolder_seg+"/Femur/femBone.Segmentation.stl".\
    format(ptNumber)
segFem = mesh.Mesh.from_file(segFempath)

# Load femoral component segmentation
segFemcompPath = ptFolder_seg+"/Femur/femComp.Segmentation.stl".\
    format(ptNumber)
segFemcomp = mesh.Mesh.from_file(segFemcompPath)

# Path to unregistered femoral component
fem_Testpath = modelFolder+"/femComp_zalign_size-{}_L{}".\
    format(FemurSideSize , LinerSize)

```

```

fem_mesh = mesh.Mesh.from_file(fem_Testpath)

# Load registered femoral component for the patient
registered_femComp = ptFolder_reg+"/Femur/femComp_Registered.stl".\
    format(ptNumber)
reg_femMesh = mesh.Mesh.from_file(registered_femComp)
,
_____

""FEMORAL COMPONENT MESH OPERATIONS""
,
_____

# Mesh reduction
""To increase the speed by at least a factor of 2, the mesh is reduced.
Realistically, all the points above the halfway point of the mesh are
not of any interest for the measurement, because only the points closest
to the tibial baseplate are required. Here, the upper half of the
original mesh is removed and the remaining data is put in a new dataset.""
reg_femMeshHalf = reduceHalf(reg_femMesh)
,
_____

""POINT GRID OPERATIONS""
,
_____

# Determine the position and size of the grid. Set the gridpoints with
# distance gridPoint_distance.
""Calculate the x and y values of the grid, the z values of the starting
grid, and the size in x and y direction. These last values are used in a
later stage for the visualization of the data.""
xGrid, yGrid, gridRange, zValues, xSize, ySize = MaxHullGrid(tib_mesh, 1,
    gridPoint_distance)

# Transform the grid points to align with the registered tibial component.
""Transform the points in xGrid and yGrid with the registration matrix
obtained from the registration. This is done by calculating the new x,
y, and z values separately.
If the transform is defined as a 4x4 matrix named M, the new values
will be:
X = x*M11 + y*M12 + z*M13 + M14
Y = x*M21 + y*M22 + z*M23 + M24
Z = x*M31 + y*M32 + z*M33 + M34
with x, y and z the original values of the datapoint.""

# Apply transform to ray start points, on the grid
for point in np.arange(xGrid.__len__()):
    transPoints = [(xGrid[point] * tib_transform[0][0]) +
        (yGrid[point] * tib_transform[0][1]) +
        (zValues * tib_transform[0][2]) + tib_transform[0][3],
        [(xGrid[point] * tib_transform[1][0]) +
        (yGrid[point] * tib_transform[1][1]) +
        (zValues * tib_transform[1][2]) + tib_transform[1][3],
        [(xGrid[point] * tib_transform[2][0]) +
        (yGrid[point] * tib_transform[2][1]) +
        (zValues * tib_transform[2][2]) + tib_transform[2][3]]]

```

```

XpointsStart.append(transPoints[0])
YpointsStart.append(transPoints[1])
ZpointsStart.append(transPoints[2])

# Apply transform to ray end points, determining the direction of the grid
for point in np.arange(xGrid.__len__()):
    transPoints = [((xGrid[point] * tib_transform[0][0]) +
                    (yGrid[point] * tib_transform[0][1]) +
                    ((zValues + 1) * tib_transform[0][2]) + tib_transform[0][3]),
                  ((xGrid[point] * tib_transform[1][0]) +
                    (yGrid[point] * tib_transform[1][1]) +
                    ((zValues + 1) * tib_transform[1][2]) + tib_transform[1][3]),
                  ((xGrid[point] * tib_transform[2][0]) +
                    (yGrid[point] * tib_transform[2][1]) +
                    ((zValues + 1) * tib_transform[2][2]) + tib_transform[2][3])]

    XpointsEnd.append(transPoints[0])
    YpointsEnd.append(transPoints[1])
    ZpointsEnd.append(transPoints[2])
,

""""MESH FACE MEASUREMENTS""""
,

#Set the mesh vectors
femVectors = reg_femMeshHalf
femVectors_x = femVectors[:, [0, 3, 6]]
femVectors_y = femVectors[:, [1, 4, 7]]
femVectors_z = femVectors[:, [2, 5, 8]]

i = XpointsStart.__len__()
nr = 0
pointNr = 0

# Calculate the normal of each face of the mesh
for face in np.arange(femVectors.__len__()):
    plane = femVectors[face]
    #
    a = [plane[3] - plane[0],
         plane[4] - plane[1],
         plane[5] - plane[2]]

    b = [plane[6] - plane[0],
         plane[7] - plane[1],
         plane[8] - plane[2]]
    #
    planeNormal = [((a[1] * b[2]) - (a[2] * b[1])),
                  -((a[0] * b[2]) - (a[2] * b[0])),
                  ((a[0] * b[1]) - (a[1] * b[0]))]

    normals.append(planeNormal)

# Calculate the distance from the starting point on the grid to each
# corresponding face using a Ray-Plane intersection

```

```

# method.
for point in np.arange(XpointsStart.__len__()):
    start_time2 = time.time()
    R0 = np.array([XpointsStart[point][0], YpointsStart[point][0],
                  ZpointsStart[point][0]])
    R1 = np.array([XpointsEnd[point][0], YpointsEnd[point][0],
                  ZpointsEnd[point][0]])

    length = []
    snijpunt = 0

    # Calculate the intersection using Ray-Plane intersection calculation
    for face in np.arange(femVectors.__len__()):
        faceNormal = normals[face]
        plane = femVectors[face]
        intersectPoint = rayplaneIntersect2(plane, R0, R1, faceNormal)
        minxBound = min(femVectors_x[face])
        maxxBound = max(femVectors_x[face])
        minyBound = min(femVectors_y[face])
        maxyBound = max(femVectors_y[face])
        minzBound = min(femVectors_z[face])
        maxzBound = max(femVectors_z[face])

        # Determine whether the intersection is within the boundaries of
        # the face. If this is the case, add the length from the ray
        # start point to the intersectPoint to the variable length.
        if intersectPoint != []:
            if intersectPoint[2] <= maxzBound and intersectPoint[2] >= \
                minzBound and intersectPoint[0] <= maxxBound \
                and intersectPoint[0] >= minxBound and \
                intersectPoint[1] <= maxyBound and intersectPoint[1] \
                >= minyBound:
                xInter.append(intersectPoint[0])
                yInter.append(intersectPoint[1])
                zInter.append(intersectPoint[2])

                linerLine = [intersectPoint[0] - R0[0],
                            intersectPoint[1] - R0[1],
                            intersectPoint[2] - R0[2]]

                linerLength = np.linalg.norm(linerLine)
                length.append(np.round(linerLength, 2))

    # Set the length of the measurement to 50 if no intersection is found.
    # Also, only keep the shortest distance. analysisLength is an extra
    # variable to provide easier analysis of the data.
    if length:
        trueLength = np.amin(length)
        analysisLength = np.amin(length)
    else:
        trueLength = 50
        analysisLength = []

    analysisData.append(analysisLength)
    pointsData.append(trueLength)

    print(i - point, "iterations to go")
    print("--- %s seconds ---" % (time.time() - start_time2), "\n")

```

```

""""SAVE THE DATA FOR FUTURE USE""""
,
np.save("Points Data Pt. 1{ }x{ }".format(ptNumber, gridPoint_distance ,
                                           gridPoint_distance), pointsData)
z = np.array(pointsData).reshape(xSize._len_(), ySize._len_())
np.save("Figure Data Pt. 1{ }x{ } - Testing".format(ptNumber,
                                                    gridPoint_distance , gridPoint_distance), z)
np.save("Analysis Data Pt. 1{ }x{ }".format(ptNumber, gridPoint_distance ,
                                           gridPoint_distance), analysisData)
,

print("--- %s seconds ---" % (time.time() - start_time))

```

12 Appendix B: Malalignment Python Code

```
"""This python script will determine the extent of malalignment of the
components of primary total knee prosthesis. To run this code, several
things are required. First, the segmentations of the femur and tibia bones.
In this code, the segmentations are expected to be in .stl format. Also,
the components are expected to be in .stl format. The registration matrix
that is referred to in this code is the registration matrix that results
from the registration in Mimics and should be in .txt format.
```

```
The loading of the data is set up to accommodate the current method of
saving data in the file and data structure employed in the accompanying
data folder. When changing this, the input values and data loading paths
should be changed accordingly."""
```

```
"""Load required packages"""
```

```
import time
start_time = time.time()

import matplotlib.pyplot as plt
from mpl_toolkits.mplot3d import Axes3D, art3d
import numpy as np
from stl import mesh #stl = numpy-stl
from scipy.spatial import ConvexHull
import scipy.misc as IM
```

```
"""DEFINE THE REQUIRED FUNCTIONS"""
```

```
"""Import the testing, femur and tibia meshes"""
```

```
def ImportData(path):
    """The function ImportData imports the mesh file and the center of
    gravity of the mesh from the given path to the
    .stl file.
```

```
Input: path: path to the .stl file of the measurement data.
```

```
Output: mesh: the mesh determine from the file path.
```

```
center of gravity: the center of gravity of the mesh file.
```

```
"""
```

```
meshFile = mesh.Mesh.from_file(path)
[volume, center_of_gravity, inertia] = \
    mesh.Mesh.get_mass_properties(meshFile)
```

```
return meshFile, center_of_gravity
```

```
"""Determine the convex hull for the femur"""
```

```
def PolyArea(x, y):
    """The function PolyArea determines the area within certain edge
    points. The area is determined by a simple
    implementation of the shoelace algorithm.
```

```
Input: x: an array of x-values of the edge points to consider
```

```
y: an array of y-values of the edge points to consider
```

Output: The area within the considered edge points."""

```
area = 0.5*np.abs(np.dot(x, np.roll(y,1))-np.dot(y, np.roll(x,1)))
```

```
return area
```

```
def HullCentroid(mesh, amount_bins):
```

```
"""Function HullCentroid determines the centroid points of the convex hulls of input mesh. For each set of points between Z[n]...Z[n+amoun_bins], a ConvexHull is determined. From that ConvexHull, the area and its centroid are calculated. This is done for each of the vertices in the mesh along the Z-direction of the global coordinate system, determined by the MRI.
```

```
Input: - mesh: the mesh for which you wish to determine the centroid line.
        - amount_bins: the step-size along the Z-direction.
                    The larger amount_bins, the more points are taken into account for determining the hull, hull area and hull centroid.
```

```
Output: - An array with all the centroids of the mesh.
```

```
"""
```

```
# Get all the z-values of the mesh and determine the unique values, # the maximum and minimum values.
```

```
Z = mesh.z
```

```
Z_new = np.unique(Z)
```

```
Z_min = Z_new.min()
```

```
Z_max = Z_new.max()
```

```
# Set up the bins for future use
```

```
start_bin = np.floor(Z_min)
```

```
end_bin = np.ceil(Z_max)
```

```
numbers = np.arange(start_bin, end_bin, amount_bins)
```

```
diff_bin = end_bin - start_bin
```

```
# Make sure that all values of Z are taken into account
```

```
if diff_bin % amount_bins != 0.0:
```

```
    last_num = end_bin + (amount_bins - diff_bin % amount_bins)
```

```
    numbers = np.append(numbers, last_num)
```

```
# Set z-coordinate of slice in the middle of numbers[i] and
```

```
# numbers[i+1]
```

```
Cent_z = []
```

```
for nums in np.arange(numbers.shape[0] - 1):
```

```
    Cent_z.append((numbers[nums] + numbers[nums + 1]) / 2)
```

```
mesh_vectors = mesh.vectors
```

```
# DETERMINE THE VARIOUS BINS CONTAINING THE VERTICES. EACH BIN
```

```
# CORRESPONDS TO A SLICE OF THICKNESS amount_bins mm.
```

```
Z_bins = []
```

```
# loop through bins
```

```
for num in np.arange(numbers.shape[0] - 1):
```

```
    Z_bin = []
```

```
    # Loop through the numbers
```

```

for Z_num in Z_new:
    # If number inside bin range put in Z_bin
    if Z_num >= numbers[num]:
        if Z_num <= numbers[num + 1]:
            Z_bin.append(Z_num)
Z_bins.append(Z_bin)

# SET EMPTY VARIABLE FOR USE IN THE NEXT PART
sizes = []
Cent_x = []
Cent_y = []

centAmount = 0
ind_number = 0
inds = []

###-----###
# This part of the loop determines the vertices that correspond to
# the slices that will be used for the measurement
# as determined above.

# FOR EACH BIN, DETERMINE THE CONVEX HULL OF ALL THE VERTICES IN
# THAT BIN.
for bin_nr in Z_bins:
    Z_temp = np.empty([1, 3])
    # SKIP EMPTY BINS
    if not bin_nr:
        inds.append(ind_number)
        ind_number += 1
        continue
    # PUT ALL RELEVANT VERTICES IN Z_temp.
    else:
        for z_num in bin_nr:
            for vector in mesh_vectors:
                if z_num in vector:
                    Z_temp = np.concatenate([Z_temp, vector], 0)

###-----###

# CALCULATE THE CONVEX HULL FOR THE BIN
Z_temporary = Z_temp
centAmount += 1
XY_temporary = Z_temporary[1:, 0:2]
hull = ConvexHull(XY_temporary)
x = XY_temporary[hull.vertices, 0]
y = XY_temporary[hull.vertices, 1]

# CALCULATE THE CONVEX HULL SURFACE AREA
sizes.append(PolyArea(x, y))

x2 = np.append(x, x[0])
y2 = np.append(y, y[0])

# Determine centroid x and y. See also:
# https://en.wikipedia.org/wiki/Centroid#CITEREFBourkeJuly\_1997
xy_mult = []
x_mult = []
y_mult = []

```

```

x_multy = []
y_multy = []

for i in np.arange(x.shape[0]):
    xy_mult.append((x2[i] * y2[i + 1]) - (x2[i + 1] * y2[i]))
    x_mult.append((x2[i] + x2[i + 1]))
    y_mult.append((y2[i] + y2[i + 1]))
    x_multy.append(xy_mult[i] * x_mult[i])
    y_multy.append(xy_mult[i] * y_mult[i])

# A = (1 / 2) * np.abs(sum(xy_mult))
A2 = PolyArea(x, y)
Cx = (1 / (6 * A2)) * sum(x_mult)
Cy = (1 / (6 * A2)) * sum(y_mult)

# PUT THE CENTROID DATA IN Cent_x AND Cent_y
Cent_x.append(Cx)
Cent_y.append(Cy)

# GO TO NEXT BIN
ind_number += 1

""""DO NOT REMOVE FROM CODE""""
'

# # Plot the vertices, hull and centroid
# if ind_number == 45:
#     fig2 = plt.figure(2)
#     plt.plot(XY_temporary[:, 0], XY_temporary[:, 1], 'o')
#     plt.plot(XY_temporary[hull.vertices, 0], XY_temporary
#             [hull.vertices, 1], 'r--', lw=2)
#     plt.plot(XY_temporary[hull.vertices, 0], XY_temporary
#             [hull.vertices, 1], 'ko', lw=2)
#     plt.plot(Cx, Cy, 'k*')
#     plt.ylim([-40, 25])
#     plt.xlim([-40, 40])
#     plt.show()
''

# REMOVE EMPTY SLICE INDICES FROM Cent_z
Cent_z = np.delete(Cent_z, inds)

# CORRECT THE DATA FOR CONVEX HULL SURFACE AREA ERRORS
correctedData = newSlices(sizes, 4)

cent_x = []
cent_y = []
cent_z = []
new_sizes = []

for i in correctedData:
    for element in i:
        cent_x.append(Cent_x[element])
        cent_y.append(Cent_y[element])
        cent_z.append(Cent_z[element])
        new_sizes.append(sizes[element])

# REMOVE THE CENTROIDS WITH X OR Y VALUES OUTSIDE OF ONE STANDARD

```

```

# DEVIATION OF THE MEAN X AND Y OF THE CENTROIDS.
I = []
for i in np.arange(cent_x.__len__()):
    if cent_x[i] <= np.mean(cent_x) - np.std(cent_x) or cent_x[i] >= \
        np.mean(cent_x) + np.std(cent_x):
        I.append(i)

    elif cent_y[i] <= np.mean(cent_y) - np.std(cent_y) or cent_y[i] \
        >= np.mean(cent_y) + np.std(cent_y):
        I.append(i)

cent_x2 = np.delete(cent_x, I)
cent_y2 = np.delete(cent_y, I)
cent_z2 = np.delete(cent_z, I)

return [cent_x2, cent_y2, cent_z2], sizes, new_sizes, correctedData

"""Fit a line to all points from previous calculation"""
def ApproxLine(line_data):
    """This function takes the 3D input data and approximates a linear
    line through the data by using Single Value
    Decomposition.

    Input: array of 3D points
    Output: start and end of the linear approximation of the 3D data.

    """

    data = np.concatenate((np.array(line_data[0])[:, np.newaxis],
                             np.array(line_data[1])[:, np.newaxis],
                             np.array(line_data[2])[:, np.newaxis]),
                           axis=1)

    XZdata = np.concatenate((np.array(line_data[0])[:, np.newaxis],
                              np.array(line_data[2])[:, np.newaxis]),
                             axis=1)

    YZdata = np.concatenate((np.array(line_data[1])[:, np.newaxis],
                              np.array(line_data[2])[:, np.newaxis]),
                             axis=1)

    # Calculate the mean of the points, i.e. the 'center' of the cloud
    datamean = np.mean(data, axis=0)
    XZdatamean = np.mean(XZdata, axis=0)
    YZdatamean = np.mean(YZdata, axis=0)

    # Do an SVD on the mean-centered data. SEE ALSO AMATH 301 SVD ON
    # YOUTUBE!!!!!!
    # Singular value decomposition determines three matrices from a
    # single input data matrix. The resulting matrices
    # give information about the correlation of the data matrix values
    # in the columns with the data matrix values in the
    # rows. Together with the sigma matrix (ss), it determines the
    # correlation of the data with itself. Thus: in our
    # case,  $A = USV^T$  comes down to  $ss[0]$  being a very large number
    # (over 300), meaning there is a strong correlation
    # of the first column of  $U$  and the first row of  $V^T$  with the data.
    # As we wish to find the correct line through the

```

```

# data, and V^T holds the resulting lines corresponding to the
# data we take vv[0] to provide the line along the
# z-axis that we wish to find. For the corresponding x-axis line
# we find vv[1], and for the y-axis we take vv[2].

u, s, v = np.linalg.svd(data-datamean)
uu, ss, vv = np.linalg.svd(XZdata-XZdatamean)
uuu, sss, vvv = np.linalg.svd(YZdata-YZdatamean)

linepts = v[0] * np.mgrid[-80:80:2j][:, np.newaxis]
linepts += datamean

XZlinepts = vv[0] * np.mgrid[-50:50:2j][:, np.newaxis]
XZlinepts += XZdatamean

YZlinepts = vvv[0] * np.mgrid[-50:50:2j][:, np.newaxis]
YZlinepts += YZdatamean

return (data, linepts), (XZdata, XZlinepts), (YZdata, YZlinepts)

"""Find unit vector"""
def UnitVec(linepts):
    """The function CreateLine determines the positively oriented unit
    vector determined from the ApproxLine. The unit
    vector has length = 1, and will always face in positive direction.
    Furthermore, the origin of the unit vector will
    be set to the center of gravity of the segmentation mesh. (This is
    fairly arbitrarily chosen, but provides an easy
    and intuitive understanding of the results. Furthermore, the
    position of the vector is not of any interest, just
    its orientation).

    Input: linepts: the 3D data points from the
                ApproxLine function above.
           center_of_gravity: the center of gravity of the input
                mesh, determined previously"""

    # Force the unit vector to face in positive z-direction.
    line_z = linepts[:, 2]
    min_z = np.argmin(line_z)
    if min_z == 1:
        line_max = linepts[0] - linepts[1]
    else:
        line_max = linepts[1] - linepts[0]

    # Determine the unit vector
    unit_vector = line_max / np.sqrt(np.square(line_max[0]) +
                                     np.square(line_max[1]) +
                                     np.square(line_max[2]))

    return unit_vector

"""Calculate angle between axes"""
def AngleCalc(vector):
    """This function determines the angle the vector makes in Z-direction
    in the ZX- and ZY-plane of the global coordinate system. The
    function takes a single 3D vector of arbitrary magnitude and
    gives the angles of interest. To determine the angles, first the

```

projection of the 3D vector on the respective plane is determined. From the projection, a simple angle measurement is performed by rewriting the dot product of two vectors, giving the angles of interest.

(<http://www.euclideanspace.com/maths/geometry/elements/plane/lineOnPlane/index.htm>)

Input: single 3D vector

Output: Angles in the ZX- and ZY-planes of the Z-direction of the input vector in radian and degrees.

First output is the angle of the vector with the Z axis of the ZX-plane of the global coordinate system.

Second output is the angle of the vector with the Z axis of the ZY-plane of the global coordinate system.

The ZX-plane is the coronal plane.

The ZY-plane is the sagittal plane.

"""

```
x_vec = [1, 0, 0]
y_vec = [0, 1, 0]
z_vec = [0, 0, 1]
```

```
ZX_norm = y_vec
```

```
ZY_norm = x_vec
```

```
plane_norm = [ZX_norm, ZY_norm]
```

```
angles = []
```

```
# HERE CALCULATE THE PROJECTION OF THE VECTOR OF INTEREST ON
```

```
# THE GLOBAL ZX- AND ZY-PLANE.
```

```
for i in np.arange(2):
```

```
    plane_projection = np.cross(plane_norm[i], np.cross(vector,
                                                         plane_norm[i])) / \
        (np.square(np.sqrt(np.square(plane_norm[i][0]) +
                                     np.square(plane_norm[i][1]) +
                                     np.square(plane_norm[i][2]))))
```

```
    inprod = z_vec[0] * plane_projection[0] + z_vec[1] * \
        plane_projection[1] + z_vec[2] * plane_projection[2]
```

```
    mag1 = np.sqrt(np.square(z_vec[0]) + np.square(z_vec[1]) +
                  np.square(z_vec[2]))
```

```
    mag2 = np.sqrt(np.square(plane_projection[0]) +
                  np.square(plane_projection[1]) +
                  np.square(plane_projection[2]))
```

```
# DETERMINE THE RELEVANT ANGLES
```

```
rad_angle = np.arccos(inprod / (mag1 * mag2))
```

```
deg_angle = (rad_angle * (180 / np.pi))
```

```
# Angle is positive if in counterclockwise direction
```

```
if plane_projection[0] > 0:
```

```
    rad_angle = -1*rad_angle
```

```

        deg_angle = -1*deg_angle
    if plane_projection[1] < 0 and plane_projection[0] < 0:
        rad_angle = -1*rad_angle
        deg_angle = -1*deg_angle

    # SET ANGLE AND ADD TO VARIABLE angles
    deg_angle = np.round(deg_angle,2)
    angles.append([rad_angle, deg_angle])

    return angles

"""Plot the information"""
def DataPlot(mesh, Line, datapoints, unit_vector, hullSizes, reducedSizes,
            center_of_gravity, reducedSlices, XZLine,
            YZLine):
    """The function DataPlot will provide several figures with the data
    from the input data.

    Input:
        mesh: the input mesh itself
        Line: the linear approximated line of the
              mesh data
        datapoints: the centroids of the ConvexHulls
        unit_vector: the unit vector of Line
        hullSizes: an array with the sizes of each
                   ConvexHull of the mesh
        center_of_gravity: the center of gravity of the input mesh"""

    # Plot the global coordinate system
    # First three is the starting point, second three is the direction
    # of the arrow/end point.
    vectors = np.array([[0, 0, 0, 1, 0, 0], [0, 0, 0, 0, 1, 0],
                       [0, 0, 0, 0, 0, 1]])

    # Provide the colors for the respective vectors
    colors = ["red", "green", "blue"]

    # Initialize the figure in 3D
    fig = plt.figure(1)
    ax = Axes3D(fig)

    itnum = 0
    for vector in vectors:
        v = np.array([vector[3], vector[4], vector[5]])
        vlength = np.linalg.norm(v)
        # Create the arrows for in the plot
        # See also https://matplotlib.org/api/\_as\_gen/matplotlib.axes.Axes.quiver.html for more
        # info on the use of ax.quiver
        ax.quiver(vector[0], vector[1], vector[2], vector[3],
                 vector[4], vector[5], length=10,
                 arrow_length_ratio=0.3 / vlength, color=colors[itnum])
        itnum += 1

    # Set axes limits, labels and plot title
    ax.set_xlabel('x')
    ax.set_ylabel('y')
    ax.set_zlabel('z')
    ax.set_title("3D Model of Bone Segmentation")

```

```

xline = [Line[0][0], Line[1][0]]
yline = [Line[0][1], Line[1][1]]
zline = [Line[0][2], Line[1][2]]

# Z = datapoints[:,2]

# Add the data to the axes
bone = art3d.Poly3DCollection(mesh.vectors)
bone.set_alpha(0.20)
bone.set_facecolor("blue")

ax.add_collection3d(bone)
ax.scatter(datapoints[:,0], datapoints[:,1], datapoints[:,2],
           facecolor='red', label="Centroid points")
ax.plot(xs=xline, ys=yline, zs=zline, label = "Determined "
        "Anatomical Axis")

# Show the data plot
fig.legend()
ax.set_xlim(-40,40)
ax.set_ylim(-40,40)
ax.set_zlim(-80, 100)
plt.show()

fig2 = plt.figure(2)
axe = fig2.add_subplot(111)
axe.plot(hullSizes, 'r-', label="Convex Hull Surface Area for "
        "each slice")
axe.plot(reducedSlices, reducedSizes, 'k--', label="Convex "
        "Hull Surface Areas "
        "for mid-shaft "
        "measurement")

fig2.legend()
axe.set_title("Surface area along the Z-axis")
axe.set_ylabel("Surface area")
axe.set_xlabel("Z-Slice")
plt.show()

"""Determine the indices of the hull to take into account for the
measurement"""
def newSlices(hullSizes, maxSize):
    """The function newSlices provides a new array with the values of
    newSlices > max/maxSize. maxSize determines the smallest numbers
    that will be used for the further measurement. If maxSize = 2, the
    Full Width at Half Maximum will be used. If maxSize = 3, the 1/3*max
    will be used as a cutoff. Default cutoff size maxSize = 4.

    Input: hullSizes: array with values of the area of each Convex Hull.
           maxSize: Value determining the extent of the mesh to be
                    used for further measurement. This determines
                    values to be used from the convex hull surface
                    area calculations.

    Output: slices: array with index values. These indices correspond
                to the bins to be used for further
                measurement."""

```

```

if not maxSize:
    maxSize = 4
else:
    maxSize = maxSize

maxPoint = np.amax(hullSizes)
cutoff = maxPoint/maxSize
slices = np.where(hullSizes >= cutoff)

return slices

"""Read transformation matrix from .txt file"""
def readTransform(path):
    """The function readTransform computes the rotation matrix form the
input .txt file containing the 4x4
transformation matrix corresponding to the registration of the
corresponding CAD model.

Input: path: the path to the .txt file containing the transformation
matrix.
Output: new Z-axis vector.
"""

# READ THE REGISTRATION TRANSFORMATION FILE
file = open(path, 'r')
data = file.readlines()

def findNR(string):
    l=[]
    tokens = string.split()
    for token in tokens:
        l.append(float(token))

    return l

# SET THE 4X4 REGISTRATION MATRIX AS MATRIX MM
M1 = findNR(data[2])
M2 = findNR(data[3])
M3 = findNR(data[4])
M4 = findNR(data[5])

MM = [M1,
      M2,
      M3,
      M4]

# CALCULATE THE SCALE FACTOR OF MM
Sx = np.linalg.norm(MM[0])
Sy = np.linalg.norm(MM[1])
Sz = np.linalg.norm(MM[2])

# CALCULATE THE ROTATION MATRIX OF MM
RotMat = [[MM[0][0] / Sx, MM[0][1] / Sx, MM[0][2] / Sx],
          [MM[1][0] / Sy, MM[1][1] / Sy, MM[1][2] / Sy],
          [MM[2][0] / Sz, MM[2][1] / Sz, MM[2][2] / Sz]]

# PROVIDE THE GLOBAL COORDINATE SYSTEM TO CALCULATE THE LOCAL
# COORDINATE SYSTEM WITH THE ROTATION MATRIX

```

```

GCS = [[1, 0, 0],
        [0, 1, 0],
        [0, 0, 1]]

# CALCULATE THE LOCAL COORDINATE SYSTEM TO DETERMINE THE NEW
# Z-AXIS OF THE COMPONENT
LCS = np.matmul(GCS, RotMat)
New_z = LCS[2]

return [New_z]

"""Determine the angle of the anatomical axis with the GCS"""
def GCS_Angle(mesh, sliceThickness, CenterOfGravity):
    # The function GCS_Angle calculates the angle of the anatomical axis
    # of the bone mesh with the GCS.
    # Input: mesh - The mesh data of the bone to be used for calculation
    #         sliceThickness - chosen sliceThickness for determining the
    #         Convex Hulls.
    #         CenterOfGravity - The cog of the mesh, for positioning of
    #         the corresponding unit vector.

    # Output: Angles - The angle of the anatomical axis of the bone with
    # the GCS.
    #-----

    # Calculate the centroids and the corrected data
    [Centroids, hullSurfaceArea, correctedSurfaceArea,
     correctedCentroids] = HullCentroid(mesh, sliceThickness)

    # Determine the line through the centroids. Also, determine the 2D
    # lines, for later visualization.
    [Line, XZLine, YZLine] = ApproxLine(Centroids)

    # Calculate the unit vector from the calculated Line from ApproxLine()
    unitVector = UnitVec(Line[1])

    # Calculate the angle of the unit vector with the global coordinate
    # system in coronal and sagittal planes. The
    # calculation utilizes projection of the 3D vector on the
    # corresponding planes to calculate the angle with the
    # plane.
    Angles = AngleCalc(unitVector)

    # Plot the data
    # DataPlot(mesh, Line[1], Line[0], unitVector, hullSurfaceArea,
    # correctedSurfaceArea, CenterOfGravity,
    # correctedCentroids[0], XZLine[1], YZLine[1])

    return(Angles)

,
,

"""!The Measurement Starts Here!"""
"""Step One: Load Data"""
,

"""Set data folder"""

```

```

#This is the HOME folder where the rest of the data is present.

# !!! THIS FOLDER SHOULD BE CHANGED WHEN A NEW/DIFFERENT FOLDER
# LOCATION IS USED !!!

folderLocation = "/Volumes/L-BOX/Afstuderen"

# This is the MAIN folder from which the rest of the data can be found.
mainFolder = folderLocation + "/Data"

# Segmentation subfolders
subFolder_Segmentation = mainFolder + "/Segmentations"
subFolder_Registration = mainFolder + "/Registrations"

# Set ptNumber
ptNumber = 7

ptNumber = str(ptNumber)
if ptNumber.__len__ < 2:
    ptNumber = "0"+ptNumber

side = "Right"

# Segmentation data is in segFolder. Except, pt.1 has 3 segmentations
# for ICC purposes. Therefore, segFolder =
# "Segmentation 1" instead of just "Segmentation". (The choice
# for the 1st segmentation is ambiguous).

if ptNumber == 1:
    segFolder = "/Segmentation 1"
    regFolder2 = "/Registration 1"
else:
    segFolder = ""
    regFolder2 = ""

# Set ptFolder to extract patient specific image data.
ptFolder_seg = subFolder_Segmentation + "/Pt. 1{}".format(
    ptNumber, segFolder)
ptFolder_reg = subFolder_Registration + "/Pt. 1{}".format(
    ptNumber, regFolder2)

""""Load the .STL files""""
# Load Tibia data
tibFolder = ptFolder_seg + "/Tibia/tibBone_Segmentation.stl"
[tibMesh, tibCog] = ImportData(tibFolder)

# Load Femur data
femFolder = ptFolder_seg + "/Femur/femBone_Segmentation.stl"
[femMesh, femCog] = ImportData(femFolder)

# Load registration matrices obtained from the Mimics Registration
# algorithm
femRegmat = ptFolder_reg + "/Femur Registration Matrix.txt"
tibRegmat = ptFolder_reg + "/Tibia Registration Matrix.txt"

""""Set the size of the slices""""

```

```

sliceSize = 1 # in mm.
,
_____,

"""Step Two: Calculate Angle of Mid-Shaft line with GCS"""
,
_____,

"""Calculate the Angle of the anatomical axis of the tibia and femur with
respect to the global coordinate system."""
# Calculate Femoral Angles
femAngles = GCS_Angle(femMesh, sliceSize, femCog)
# Calculate Tibial Angles
tibAngles = GCS_Angle(tibMesh, sliceSize, tibCog)
,
_____,

"""Step Three: Calculate the Angle of the Prosthesis Coordinate System
with the GCS"""
,
_____,

"""To calculate the angle of the prosthesis local coordinate system the
registration matrix from the registration in
Mimics is loaded. The registration matrix holds the transformation of the
3D model onto the segmentation.
Because the 3D model is aligned with the GCS, applying the same
transformation to the GCS will yield the LCS of
the prosthesis."""

# Locate and read registration matrices"""
femComp_Z = readTransform(femRegmat)
tibComp_Z = readTransform(tibRegmat)

# Calculate angle of prosthesis z-axis with GCS.
fem_compAngles = AngleCalc(femComp_Z[0])
tib_compAngles = AngleCalc(tibComp_Z[0])
,
_____,

"""Step Four: Calculate the relevant angles for Malalignment"""
,
_____,

"""Femoral angles"""
# Correct for left or right side
if side == "Left":
    cFA = 90 + np.round((fem_compAngles[0][1] - femAngles[0][1]), 2)
else:
    cFA = 90 - np.round((fem_compAngles[0][1] - femAngles[0][1]), 2)

sFA = np.round((fem_compAngles[1][1] - femAngles[1][1]), 2)

if cFA < -7:
    print("The coronal Femoral Angle (cFA) = ", np.abs(cFA),
          " degrees valgus")
elif cFA > 3:
    print("The coronal Femoral Angle (cFA) = ", np.abs(cFA),
          " degrees varus")

```

```

else:
    print("The coronal Femoral Angle (cFA) = {} degrees. "
          "Within expected range".format(cFA))

if sFA < 0:
    print("The sagittal Femoral Angle (sFA) = {} "
          "degrees retroversion".format(sFA))
elif sFA > 0:
    print("The sagittal Femoral Angle (sFA) = {} "
          "degrees anteversion".format(sFA))
else:
    print("The sagittal Femoral Angle (sFA) = 0. "
          "Perfect alignment with the midshaft of the femur.")

"""Tibial angles"""
# Correct for left or right side
if side == "Left":
    cTA = 90 - np.round((tib_compAngles[0][1] - tibAngles[0][1]), 2)
else:
    cTA = 90 + np.round((tib_compAngles[0][1] - tibAngles[0][1]), 2)
sTA = np.round((tib_compAngles[1][1] - tibAngles[1][1]), 2)
# print(cTA, sTA)

if cTA < 180:
    print("The coronal Tibial Angle (cTA) = ", np.abs(cTA),
          " degrees valgus")
elif cTA > 180:
    print("The coronal Tibial Angle (cTA) = ", np.abs(cTA),
          " degrees varus")
else:
    print("The coronal Tibial Angle (cTA) = {} degrees. "
          "Within expected range".format(cTA))

if sTA > 180:
    print("The sagittal Tibial Angle (sTA) = {} "
          "degrees retroversion".format(np.abs(sTA)))
elif sTA < 180:
    print("The sagittal Tibial Angle (sTA) = {} "
          "degrees anteversion".format(np.abs(sTA)))
else:
    print("The sagittal Tibial Angle (sTA) = 0. "
          "Perfect alignment with the midshaft of the tibia.")

truecTA = 90+cTA

TFA = femAngles[0][1] + tibAngles[0][1]
TFA2 = cFA + cTA
print("The Tibio-Femoral Angle (TFA) = ", TFA, ". This is the angle "
      "between the anatomical axes of the tibial and femoral bones.")
,
,
,
print("--- %s seconds ---" % (time.time() - start_time))
,

```

```
""""SAVE THE DATA""""
```

```
,
```

```
np.save("cFA Pt. 1{}".format(ptNumber), cFA)
```

```
np.save("cTA Pt. 1{}".format(ptNumber), cTA)
```

```
np.save("sFA Pt. 1{}".format(ptNumber), sFA)
```

```
np.save("sTA Pt. 1{}".format(ptNumber), sTA)
```

```
np.save("TFA Pt. 1{}".format(ptNumber), TFA)
```

```
""""End of Measurement""""
```

```
,
```