



TÉCNICO
LISBOA

3D Statistical Shape Model of Proximal Femur for Femoroacetabular Impingement Preoperative Planning

Rafael Lampreia dos Santos Fataça

Thesis to obtain the Master of Science Degree in

Mechanical Engineering

Supervisor: Prof. Jorge Manuel Mateus Martins

Examination Committee

Chairperson: Prof. Carlos Baptista Cardeira

Supervisor: Prof. Jorge Manuel Mateus Martins

Member of the Committee: Prof. Mário António Da Silva Neves Ramalho

December 2021

Declaration

I declare that this document is an original work of my own authorship and that it fulfills all the requirements of the Code of Conduct and Good Practices of the Universidade de Lisboa.

Acknowledgments

First and foremost, I would like to express my gratitude to my supervisor, Prof. Jorge Martins, for his guidance during the development of my master's thesis in a field I personally find specially fascinating, orthopaedic surgery. I would also like to thank João Oliveira, from the Surgical Robotics Lab, for all the constant support, patience, ideas and going the extra mile to explain the Curan project architecture to someone that had no background of C++.

To all the friends I made during my time in Técnico, specially the ones from the student association groups I was part of, CPMEMec, Fórum Mecânica and AEIST, and from the Erasmus experience, a big thank you for all the opportunities and fun times, without which my academic experience would have been much blander.

To the special ones, for all the time we spent together during this journey, for all the nights in Abarde and for the uncountable memories here and across borders, a huge thank you. It is thanks to you that it feels like time flew so fast.

To all my friends from Mem Martins and Sintra, thank you for your immense support in another chapter of my life and for being there for me for such a long time, specially during the pandemic.

To my boo, thank you for all your support, patience and for always being there for me. Thank you for helping me being a better version of myself.

Last but, surely, not least, I would like to thank my family, specially my parents for the unconditional love and for always giving me the tools to build my future. Thank you for making me the person I am today and for the continuous support throughout my whole life. I am proud of being your son.

Resumo

Atualmente, a cirurgia ortopédica é auxiliada por software que ajuda tanto no diagnóstico como no planeamento préoperatório cirúrgico. O Curan tem o objetivo de combinar ambas as vertentes num software open-source, com funcionalidades de visualização e comunicação com periféricos, tais como sistemas robóticos. Este trabalho apresenta como obter e usar um modelo estatístico de forma 3D (SSM) do fémur proximal como ferramenta de planeamento pré-cirúrgico para detetar danos causados pelo conflito femoroacetabular (CFA) do tipo cam. O projeto Curan é apresentado e a implementação de uma janela de visualização de imagens médicas com uma ferramenta de segmentação manual é explicada. O SSM 3D do fémur proximal é obtido através da base de dados de 33 fémures do projeto de investigação de osteoporose VPHOP. O registo foi efetuado através dos algoritmos de Análise de Componentes Principais (PCA) e de Iterative Closest Point (ICP), o último com um erro médio na ordem de 0.0387 mm. Os modos de variação do SSM foram também estudados num esforço de perceber a influência de cada vetor próprio na variação dos dados, resultando que os primeiros 26 vetores próprios representam 90.57% da variação na base de dados e 30 vetores próprios representam 97.61%. O modelo estatístico foi então usado para obter uma visualização da lesão do tipo cam num fémur patológico obtido através de segmentação manual de imagens de ressonância magnética (MRI) radiais. Finalmente, este trabalho é concluído com uma revisão global e discussão dos resultados obtidos, e sugestões para investigação futura no âmbito deste projeto.

Palavras-chave: Planeamento Pré-Operatório Cirúrgico, Statistical Shape Model (SSM), Conflito Femoro-acetabular (CFA), Registo, VPHOP

Abstract

Nowadays, orthopaedic surgeries are heavily aided by technology, particularly by software tools that help both in diagnosis and preoperative planning. Curan aims to be a software tool that combines both into the same application, by allowing for medical imaging visualization of clinical studies and communication with external peripheral, such as robotic systems. The present work presents how a 3D Statistical Shape Model (SSM) of the proximal femur can be obtained and used as a preoperative planning tool in Curan to detect cam type femoroacetabular impingement (FAI). The Curan project is presented and the implementation of a medical visualization window with a manual segmentation tool is explained. The 3D SSM of a proximal femur was obtained from a 33 femora database of an osteoporotic research project, VPHOP. Registration was performed using Principal Component Analysis (PCA) and Iterative Closest Points (ICP) algorithms, the latter having a root mean square error in the order of 0.0387 mm. The SSM modes of variation were also studied to understand the relevancy of each eigenmode for the variation in the training set with a result of 26 eigenmodes describing 90.57% of the variation and 30 eigenmodes describing 97.61%. Then, the obtained SSM was used to obtain a visual representation of the surface representing cam type lesion in a pathological femur from an FAI patient obtained by manual segmentation of radial MRI sequences. To conclude, the results are discussed and suggestions for future work are given, specially in a way of improving the present model.

Keywords: Preoperative Surgical Planning, Statistical Shape Model (SSM), Femoroacetabular Impingement (FAI), Registration, VPHOP

Contents

Declaration	iii
Acknowledgments	v
Resumo	vii
Abstract	ix
List of Tables	xiii
List of Figures	xv
List of Acronyms	xvii
1 Introduction	1
1.1 Motivation	1
1.2 Objectives	2
1.3 Thesis Outline	2
2 Background	3
2.1 Femur and Hip Joint Anatomy	3
2.1.1 Femur Anatomy	3
2.1.2 Hip Joint Anatomy	4
2.2 DICOM Files	4
2.3 Femoroacetabular Impingement (FAI)	5
2.3.1 Definition	5
2.3.2 Diagnosis	6
2.3.3 Cam Type FAI	10
2.3.4 Pincer Type FAI	10
2.3.5 Treatment	11
2.4 State of the Art	13
2.4.1 3D Slicer	13
2.4.2 OsiriX	14
2.4.3 DYONICS PLAN	14
2.5 Surgical Robotics Lab and previous work	15
3 Curan Software Development	17
3.1 Build and API	17

3.2	FAI Preoperative Planning	19
3.2.1	Medical Imaging Visualizer Implementation	19
3.2.2	Manual Segmentation Options Box	21
4	Statistical Shape Model	23
4.1	Algorithm	23
4.2	Training Set	25
4.3	Point Registration	26
4.3.1	Principal Component Analysis	26
4.3.2	Data Refinement	28
4.3.3	Iterative Closest Point	29
4.4	Point Remapping and Mean Shape model	30
4.5	Pathological Femur model	32
4.6	FAI Identification	33
5	Results and Discussion	35
5.1	Mean Shape femur	35
5.1.1	PCA registration	35
5.1.2	ICP registration	38
5.1.3	Mean Shape model	39
5.2	Statistical Shape model	41
5.3	Pathological Model and FAI Identification	44
6	Conclusions	51
6.1	Future Work	52
	Bibliography	53
A	FemurID Table	57

List of Tables

A.1 Information from each femur in the database	58
---	----

List of Figures

2.1	Anterior and posterior view of the right human femur [3]	3
2.2	Hip joint anatomy [6]	4
2.3	Exercises performed during a physical examination	6
2.4	Hip impingement test FADIR (on left) and FABER (on right) [15]	7
2.5	Anterior (a) and Dunn's 45° view (b) on a bilateral FAI patient [16]	7
2.6	Computerised Tomography scan from a patient with cam type FAI [17]	7
2.7	Imaging planes exemplified as slices of an orange [18]	8
2.8	Arrangement of radial MR-sequences and how they are distributed around the femoral head with 10° difference	9
2.9	Artifact presence in two radial MRI slices from the radial MRI series	9
2.10	Cam type FAI and location of the bone deformation [11]	10
2.11	Pincer type FAI with anterior acetabular retroversion (a) with the anterior acetabular rim (white dotted line) protruding over the posterior acetabular (black dotted line) rim (b) [11]	11
2.12	Hip arthroscopy simulation (a) and on a real patient (b) [21]	11
2.13	Surgical hip dislocation procedure [22]	12
2.14	Mini-direct anterior approach [24]	12
2.15	3DSlicer layout [26]	13
2.16	OsiriX layout [28]	14
2.17	DYONICS PLAN layout [30]	15
2.18	Difference between pathological femur (blue) and statistical model (green) shown in red [2]	15
3.1	FAI Visualizer in Curan	20
3.2	Steps of drawing a contour in Curan	21
4.1	Thirty-two point model of the boundary of a resistor from [41]	24
4.2	Femur FEM01098 from VPHOP database in ANSYS Mechanical	25
4.3	Meshes from three different femora in the database	26
4.4	Original left femur FEM01239 and after reflection by the xOy plane	27
4.5	Principal components plotted on respective femur	28
4.6	Anterior view of two femora with different shaft angle	28
4.7	Point cloud and mesh of the femoral head and neck mean shape	31

4.8	Manual segmentation of a radial slice in MATLAB for $N = 130$	32
4.9	Pathological femur representation in space	33
5.1	Result of registration using PCA	36
5.2	Result of registration using PCA	36
5.3	PCA registration of femora FEM01407 and FEM01462 before and after shaft removal	37
5.4	Shift of some femora in the point cloud as a result of the shaft reduction	37
5.5	Mean RMSE of each femur if used as reference	38
5.6	Resulting point clouds from ICP registration	39
5.7	Evolution of number of points of femur FEM01388 based on grid step value	40
5.8	Posterior view of femur FEM01543 point cloud before (a) and after downsampling (b)	40
5.9	Obtained mean shape of the femur	41
5.10	Individual and cumulative variance of eigenvalues	42
5.11	First 3 modes of variation of the SSM of the femur in posterior coronal (left column) and axial views (right column)	43
5.12	Root mean square error variation based on the number of points per radial MRI slice	44
5.13	Result from PCA registration	45
5.14	Result from ICP Registration	45
5.15	Reduction of the mean shape femur to the most relevant areas for the presented FAI femur	46
5.16	Exemplification of how the remapping algorithm is susceptible to changes in point distribution	47
5.17	Some views of the points at a distance greater than x_{up} ((a) and (b)) and below x_{low} ((c) and (d))	48
5.18	Some views of the detected region to be removed from the FAI femur	48
5.19	Some views of the top and bottom limits of the region to remove from the FAI femur	49

List of Acronyms

2D Two Dimensional.

3D Three Dimensional.

API Application Programming Interface.

CT Computed Tomography.

DICOM Digital Imaging and Communications in Medicine.

FAI Femoroacetabular Impingement.

ICP Iterative Closest Point.

IGT Image Guided Therapy.

ITK Insight Toolkit.

MRI Magnetic Resonance Imaging.

PCA Principal Components Analysis.

PDM Point Distribution Model.

RMSE Root Mean Square Error.

ROI Region of Interest.

SRL Surgical Robotics Lab.

SSM Statistical Shape Model.

SVD Singular Value Decomposition.

TCP Transmission Control Protocol.

UDP User Datagram Protocol.

VPHOP Osteoporotic Virtual Physiological Human Project.

Chapter 1

Introduction

1.1 Motivation

Orthopaedic surgeries' success, as well as most surgical procedures, is heavily dependent on the preoperative planning. Preoperative planning involves the revision of the techniques to be performed and write a step by step draft of the surgical procedure. Historically, this involved methods such as using copies of x-ray sheets in order to practice and explain to the surgical team what was the course of action [1]. Nowadays, depending on the medical specialty, there are numerous tools to help the preparation of a surgery, most of them associated with software capable of interpreting modern medical imaging techniques, such as CT or MRI scans.

Typically, medical imaging software are tools that help a health professional, or researcher, visualizing a set of images describing a certain pathology with functionalities that allow a better understanding of the clinical case, either by isolating a certain region of interest (segmentation), annotating or describing anomalies by, e.g., measuring distances. These software are often closed source and owned by medical companies, not allowing tool development by researchers for specific case studies and often being quite expensive to be used for scientific research.

Given the necessity of having an open source medical visualization tool that also enables communication with external peripherals, such as robotic systems, the Curan project was initiated in the Surgical Robotics Lab (SRL), in IST. The goal is to offer an "all in one" software, totally built in C++ with state of the art libraries such as ITK or Skia. Curan is currently in development but aims to feature previously mentioned tools, as well as additional functionalities regarding medical applications.

In the SRL, previous work regarding preoperative tools for orthopaedic surgery planning, had already been developed, more specifically, a 2D SSM of the human proximal femur that aimed to help planning surgery for cam type FAI [2]. Cam type FAI is a medical condition where the femoral head-neck junction has an abnormal bone growth creating an irregular geometry causing mechanical collision between the femoral head and the acetabulum.

An SSM is a statistical analysis of an object's shape capable of defining a set of objects by using their mean shape and a number of modes of variation. This statistical approach will be used for the cam type

FAI since femora suffering from this pathology will present abnormal geometry that can be identified when comparing to the SSM. In this thesis, a 3D SSM will be developed in an attempt of mathematically defining cam type FAI lesions on a pathological femur to be later implemented as a functionality in Curan.

1.2 Objectives

The scope of this thesis is to create a 3D Statistical Shape Model of the human proximal femur to be used in Curan as a preoperative planning tool for cam type FAI, as well as the implementation of segmentation algorithms in the Curan software. For this, a database of healthy femora and MRI radial images sequence of a patient with cam type FAI were used.

This thesis can be divided into the following objectives:

- Understanding and help building the underlying architecture of the Curan project and implement a manual segmentation tool;
- Data analysis and treatment to obtain a statistical shape model of the proximal femur from a database;
- Segmentation of a femur from a cam type FAI patient;
- Analysis of the comparison between the SSM and the pathological femora.

1.3 Thesis Outline

This thesis is divided into six chapters. The current one is an introductory chapter explaining the current thesis' motivation and its structure. The second chapter, Background, presents some of the background knowledge that will later be used into the development of the thesis, such as anatomy concepts or medical data. In this chapter, the state of the art of medical image visualization tools is also presented in order to contextualize the Curan project as well as previous work developed in the SRL related to both SSM and cam type FAI. Chapter 3, Curan, introduces the Curan project and explains its structure. It also presents the current development of the FAI Preoperative Planning tool and the implementation of a segmentation algorithm. Chapter 4, Statistical Shape Model, presents the methods and materials used for the development of the SSM of the proximal femur, the acquisition of the pathological femur and the FAI identification by comparing both models. In chapter 5, Results and Discussion, the results obtained are presented and a thorough analysis and discussion is performed. Lastly, the chapter 6, Conclusion, provides an overview of the work presented, both findings and limitations, as well as suggestions for future work.

Chapter 2

Background

2.1 Femur and Hip Joint Anatomy

2.1.1 Femur Anatomy

The human femur is the largest bone in the human body and the only bone in the thigh, being situated between the hip and knee joints. Having a great number of tendons and muscles attached to it, it is responsible for the upright natural position of humans and their ambulation. It is divided into three distinct parts, the proximal femur, shaft and distal femur. This thesis will focus on the proximal femur since it is in this region that the femoroacetabular impingement will occur. The proximal femur is composed by the head, neck and two bony processes called greater and lesser trochanters. There are two bony ridges that connect these trochanters, one on the anterior side - trochanteric line - and one on the posterior one - trochanteric crest (figure 2.1).

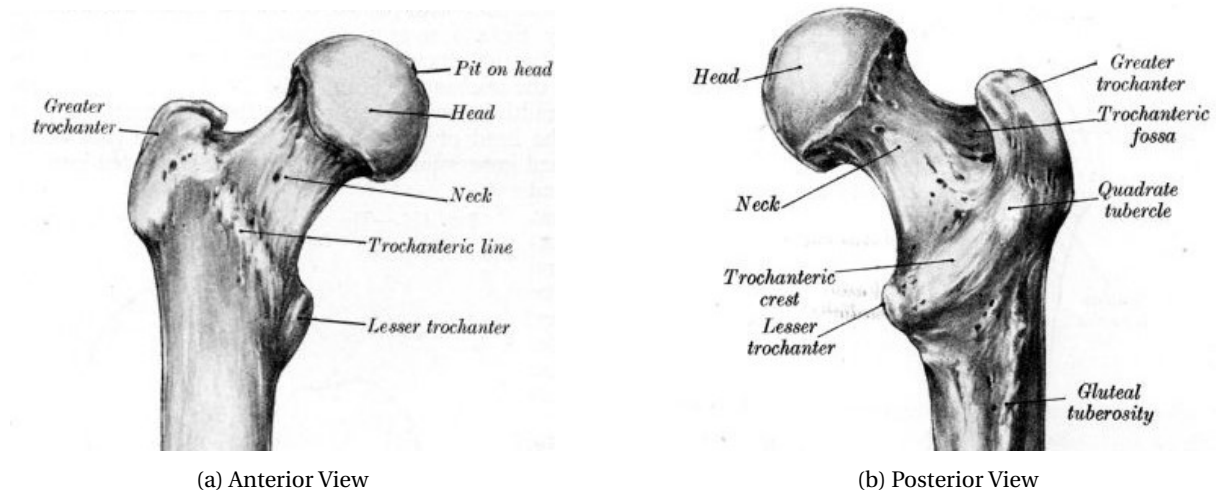


Figure 2.1: Anterior and posterior view of the right human femur [3]

The head of the femur has an almost spherical shape and it connects the femur to the acetabulum to form the hip ball and socket joint, allowing for a broad range of movement. The neck of the femur has a cylindrical shape and connects the head to the shaft at angle around 135°. This angle is also responsible to

the greater freedom of movement. Both trochanters are projections of bone responsible for holding most of the muscles and ligaments around the pelvis area. The greater trochanter has attached most of the muscles of the gluteous area while the lesser trochanter has the iliopsoas muscle attached to it [4].

2.1.2 Hip Joint Anatomy

The hip joint is one of the most important joints in the human body. This ball and socket joint is responsible for the stability and weight bearing of the whole body as well as ambulation [5].

The hip joint is an articulation connecting the femur to the pelvis in a ball and socket anatomy. The femoral head is the ball and the acetabulum the socket. The femoral head is kept inside the acetabulum due to a strong ligament called *ligamentum capitis femoris*, or ligament of head of femur. Both the head of femur and the acetabulum are covered in articular cartilage, which allows for movement without bone damage and, this cartilage, is thicker at higher stress and weight-bearing zones. The acetabular cavity is deepened by the existence of a collar around the bone called acetabular labrum. The anatomy of the hip joint is shown in figure 2.2.

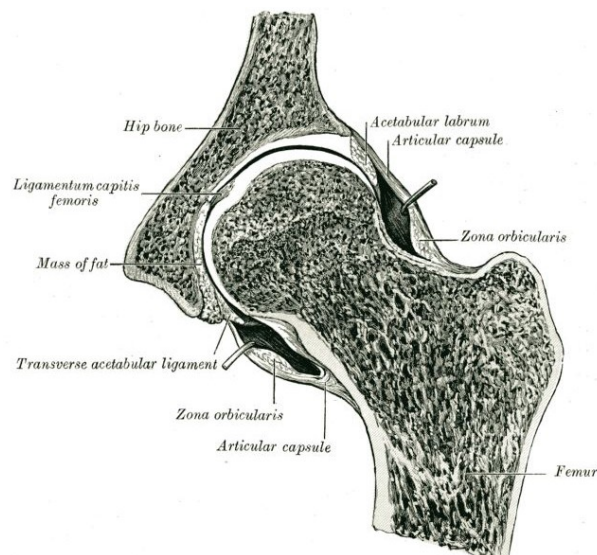


Figure 2.2: Hip joint anatomy [6]

2.2 DICOM Files

Digital Imaging and Communications in Medicine (DICOM) files are the standardized image file format used by medical imaging hardware devices. DICOM files not only store the digital imaging but also accessible data of a patient or about the image acquisition protocol [7]. In DICOM files, the data is encoded into specific unique tags, each containing different types of information, depending on the used imaging technique or procedure.

In this work, the MRI image files of the femur with cam type femoroacetabular impingement are DICOM files. Given the goal of segmenting the femur it is important to convert the coordinates of each pixel in each

MRI slice to the real 3D coordinate system. In each DICOM file there is information that helps on achieving this, more specifically [8],

- Image Position (Patient), with tag (0020,0032), is the (x, y, z) coordinates of the upper left corner of the image in mm, with respect to patient's orientation;
- Image Orientation (Patient), with tag (0020,0037), is the direction cosines of the first row and the first column with respect to the patient;
- Pixel Spacing, with tag (0028,0030), has two values that represent the physical distance in the patient between the center of each pixel in the row direction (first value) and in the column direction (second) in mm.

From [9], it is possible to obtain the real coordinates, P_{xyz} , from the (i, j) pixel by using,

$$\begin{bmatrix} P_x \\ P_y \\ P_z \\ 1 \end{bmatrix} = \begin{bmatrix} X_x \cdot \Delta_i & Y_x \cdot \Delta_j & 0 & S_x \\ X_y \cdot \Delta_i & Y_y \cdot \Delta_j & 0 & S_y \\ X_z \cdot \Delta_i & Y_z \cdot \Delta_j & 0 & S_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} i \\ j \\ 0 \\ 1 \end{bmatrix} \quad (2.1)$$

where P_{xyz} are the coordinates of voxel (i, j) in the frame's image plane in mm, S_{xyz} are the three values of the Image Position (Patient), X_{xyz} are the values from the row direction cosine of Image Orientation (Patient), Y_{xyz} are the values from the column direction cosine of Image Orientation (Patient), Δ_i is the column pixel resolution of Pixel Spacing, Δ_j is the row pixel resolution of Pixel Spacing and (i, j) are the column and row indexes, respectively, to the image plane.

2.3 Femoroacetabular Impingement (FAI)

2.3.1 Definition

Femoroacetabular impingement (FAI) is a pathomechanical condition that affects the normal functioning of the human hip joint causing abnormal contact stresses in this area [10]. In most cases, FAI is caused by a deformation in the bone of the femoral head or head/neck junction, acetabulum or both. Due to these deformations in the bone, repetitive movement of the femur in the acetabulum may lead to chondral lesions (degradation of joint cartilage) and labral tears that can eventually cause early degenerative disease [11].

FAI is a recently proposed medical term, being used for the first time in 1999 even though evidence shows records of typical FAI deformations throughout literature [12]. It started being looked upon the scientific community after studies emerged suggesting that FAI could be a cause of osteoarthritis of the hip and successful medical interventions were made in adolescents and active adults with imaging evidence of FAI. It has, since then, become important to develop assessment and treatment standards in order to improve the quality of diagnosis, as well as reducing the rate of osteoarthritis caused by this condition [10].

Clinical evidence of femoroacetabular impingement is characterized by anterior inguinal pain and decreased range of motion with flexion, adduction and internal rotation. In terms of classification, it is con-

sidered to exist two types of femoroacetabular impingement, cam type and pincer type that can coexist in a combined form.

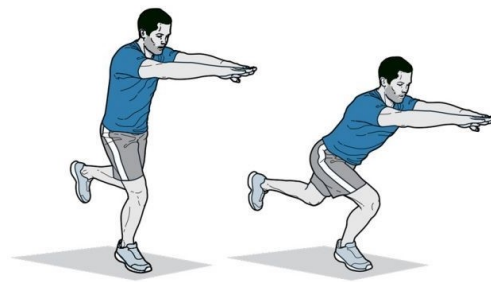
2.3.2 Diagnosis

The diagnosis of FAI consists of a combination of symptoms, physical examination and imaging studies. With this information it is then possible to distinguish FAI from intra- and extra-articular hip disorders. Even though FAI is commonly associated with groin pain and decreased range of motion, one can live years with this condition without knowing. Most FAI patients are typically young and active, usually practicing sports that require repetitive hip movement. During physical activity it is possible for the individual to feel stiffness, pain or popping of the hip joint in specific movements like joint loading, rotation or hip flexion.

In a physical examination to determine the existence of FAI, a professional will perform a set of movements in order to track down the origin of the patient's problem. Exercises like a single-leg stance (figure 2.3a) or single-leg squat (figure 2.3b) can help the professional understanding which type of movements provoke what type of pain. The physical examiner can also observe the patient's sitting posture since FAI can also affect the patient in this position.



(a) Single leg stance position [13]



(b) Single leg squat movement [14]

Figure 2.3: Exercises performed during a physical examination

The physical examiner proceeds to perform the hip impingement test on the patient. With the hip and knee flexed on a 90° angle, the hip is rotated from external rotation to internal rotation while moving from abduction to adduction (FABER - flexion, abduction and external rotation - to FADIR - flexion adduction and internal rotation, figure 2.4). Patients with a positive impingement test will feel pain at any given moment, that usually matches with the symptoms. Over 90% of the patients with confirmed FAI by radiology or other imaging techniques have a positive impingement test.

Even if a patient comes out having a positive hip impingement test it is usually necessary to have more proof of the existing problem and, if the FAI is indeed confirmed, the doctor should have more detail regarding that patient's case. For this plain radiology, computerised tomography (CT) or magnetic resonance imaging (MRI) can be used.

An x-ray is usually taken in order to confirm the physical examiner diagnose since it is the cheapest, fastest and more widely available option. It is important that the x-ray is centered on the pubic symphysis and that there is a neutral tilt between the machine and the patient. Apart from a frontal view (figure 2.5a) it is also important to have lateral hip projections such as Dunn's 45° view (figure 2.5b).

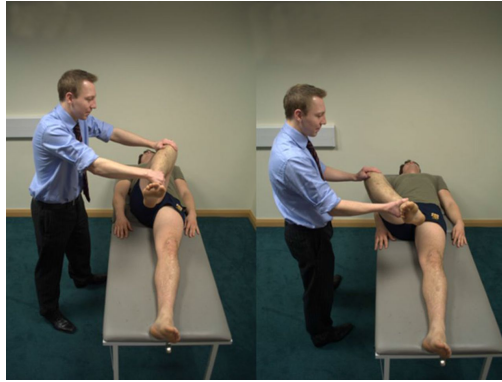


Figure 2.4: Hip impingement test FADIR (on left) and FABER (on right) [15]



(a) Anterior view



(b) Dunn's 45° view

Figure 2.5: Anterior (a) and Dunn's 45° view (b) on a bilateral FAI patient [16]

Using computerised tomography it is possible to observe osseous morphological abnormalities of the acetabulum and femoral head-neck junction allowing the doctor to identify the type of FAI in hands. With CT scans it is also possible to obtain a 3D reconstruction in order to have a better surgical planning.



Figure 2.6: Computerised Tomography scan from a patient with cam type FAI [17]

Magnetic Resonance Imaging (MRI) can be used to also detect the extent of osseous abnormalities in the hip joint in order to detect the FAI type of the patient. Since the MRI also shows cartilage it is also important

to note that, with this method, the doctor can comprehend the extent of cartilage damage such as chondral injuries like chondrolabral separation and carpet lesion. Similar to CT, MRI also enables the option of 3D reconstruction if the doctors feel like it is needed.

MRI is widely used for surgical planning of FAI since it gives the option of having radial MR-sequences. Radial sections provide a true cross section of the hip joint anatomy, allowing to have better perception of the region of interest. With conventional imaging planes, the images can be subject to averaging effects that can alter the amount of cartilage and labral damage due to the complex shape and anatomic position. The spherical shape of the femoral head and the oblique anatomic position of the head-neck joint are the primary difficulties found when using conventional imaging planes. Furthermore, in axial and coronal planes, the regions most affected by these partial-volume averaging effects are the antero- and posterosuperior regions of the hip joint while in sagittal plane, the most affected area is the superior. Radial scans differ from conventional ones since all planes go through the center of the joint, allowing for same-sized sections that contain the right proportion of information. Figures 2.7a and 2.7b illustrate how conventional and radial imaging planes differ in terms of obtainable information when the object in study has a round shape [18]. Figure 2.8, shows an example on how a radial MR-sequence is obtained [19].

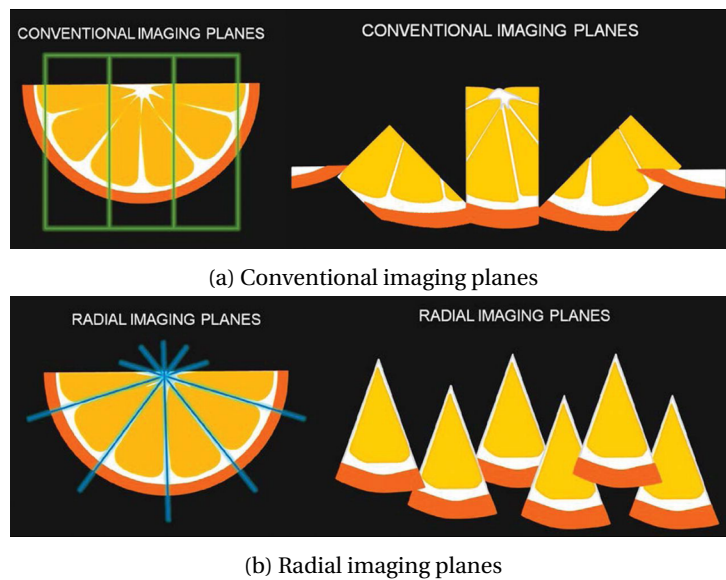


Figure 2.7: Imaging planes exemplified as slices of an orange [18]

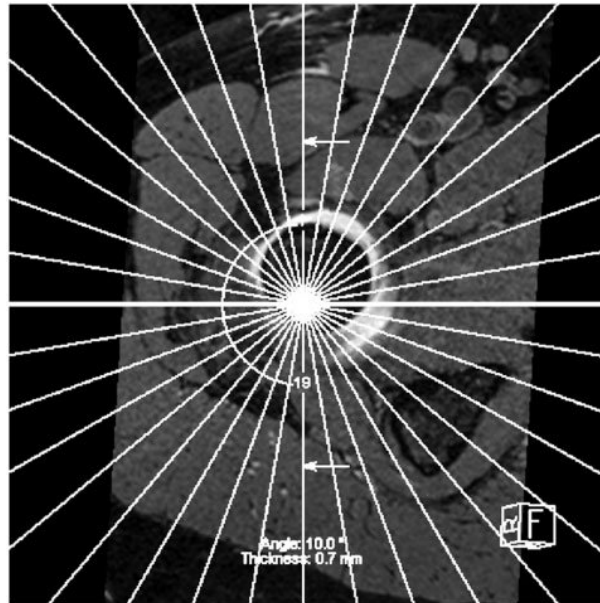
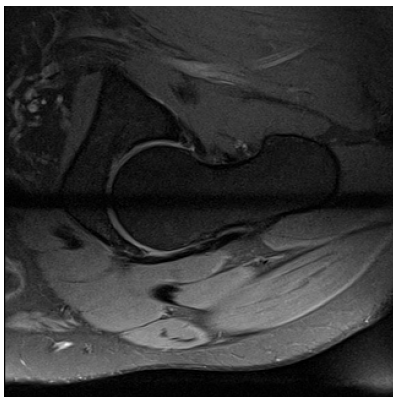
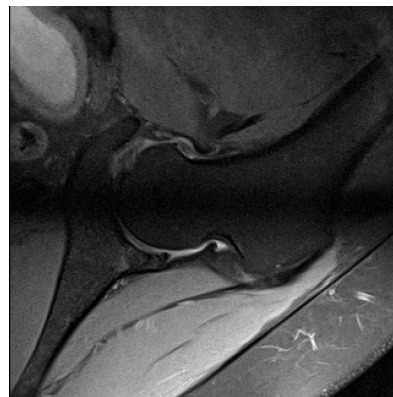


Figure 2.8: Arrangement of radial MR-sequences and how they are distributed around the femoral head with 10° difference

Unfortunately, one of the major drawbacks from radial MRI images is the inability of correctly capturing the region near the axis of revolution creating a black band in the middle of images, a visual artifact. This anomaly darkens the center of all images in a radial series affecting the quality of image in this area. In figure 2.9 it is possible to visualize the artifact present in two images from the DICOM files with the MRI images of a cam type FAI patient that will be used later to obtain the pathological femur model.



(a) Image 9



(b) Image 19

Figure 2.9: Artifact presence in two radial MRI slices from the radial MRI series

2.3.3 Cam Type FAI

In cam type impingement of the hip, the main deformation occurs in the contour of the anterior/superior femoral head-neck junction while the acetabulum has a normal morphology. A healthy femoral head-neck junction has a concave morphology (as seen in figure 2.2 between the femur and the femoral head). When a cam type femoroacetabular impingement exists, it affects the shape of this junction to a flat or, even, convex one (figure 2.10). This type of FAI can also compromise the sphericity of the femoral head.



Figure 2.10: Cam type FAI and location of the bone deformation [11]

During vigorous and repetitive movements, like the ones performed during physical activity, the non-spherical part of the femoral head will be forced against the acetabular rim, will result in damage to the existing cartilage. Cam type FAI is known by extensive cartilage damage and labral tears. These impairing lesions are more severe than the ones presented by pincer type FAI since the aspherical shape of the femoral head heavily compresses the joint with rotational and flexing movements of the leg.

2.3.4 Pincer Type FAI

The pincer type femoroacetabular impingement is characterized by the atypical morphology of the acetabulum while having a normal-shaped femur. In this case the malformations in the acetabulum can be of several types such as acetabular retroversion, anterior and/or lateral over-coverage and protrusio acetabulae. The first is shown in figure 2.11. Without diving too deep in this type of FAI, since the main focus of this thesis is on the cam type one, all these conditions affect the overall geometry around the femoral head and acetabulum by increasing the relative depth of the latter or an excessive bone growth covering the femoral head, restricting its movement in the socket.

Continuous abrasion from the femoral neck against the altered acetabular rim leads to both damage in the rim as well as the appearance of a pincer abutment. Usually, degeneration of the labrum occurs and an intra-substance cyst or ossification occurs, further worsening the problem of over-coverage of the femoral head. In pincer type FAI, acetabular chondral lesions occur in a smaller area of the acetabular rim, when compared to the cam type FAI, therefore being less impairing.

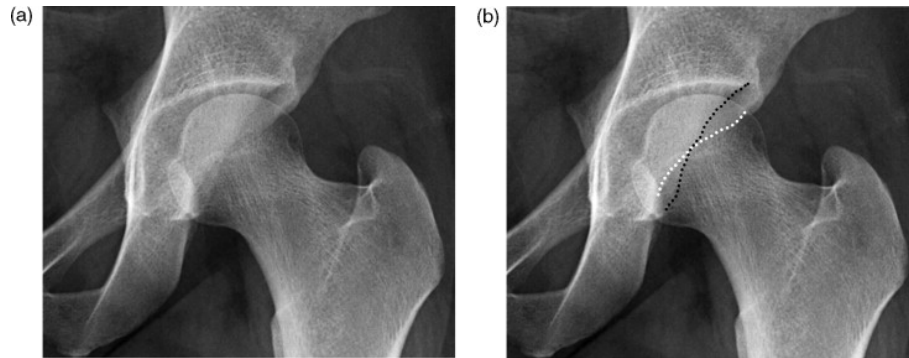
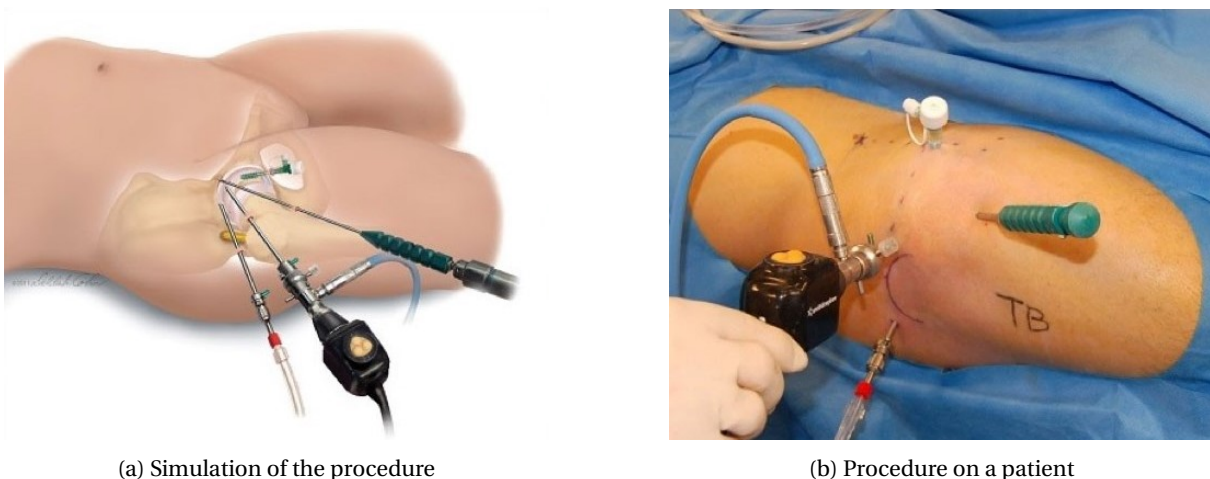


Figure 2.11: Pincer type FAI with anterior acetabular retroversion (a) with the anterior acetabular rim (white dotted line) protruding over the posterior acetabular (black dotted line) rim (b) [11]

2.3.5 Treatment

Currently, when justifiable, FAI is treated by performing a type of surgery called Femoroacetabular Osteoplasty (FAO) where the main goal is to remove excess bony areas formed on the hip bone and/or the femoral head-neck junction, reshaping these areas in order to minimize the abrasion between both parts. It may also be needed to repair torn labrum. To do so, one of three types of surgery may be used to correct such damage [20].

The first possible type of surgery is Hip Arthroscopy. Arthroscopy is a minimally invasive type of surgery where the area to be operated is accessed through small incisions making the recovery easier on the patient. Through these incisions an arthroscope, a small high quality videocamera, is inserted to observe the joint while the tools are inserted on the remaining incisions to perform the surgery. Usually, the typical damage caused by FAI, such as labral tear, damaged articular cartilage and bone spurs, can be treated using this type of surgery. Hip arthroscopy, usually, takes no longer than two hours depending on the extension of damage caused by FAI.



(a) Simulation of the procedure

(b) Procedure on a patient

Figure 2.12: Hip arthroscopy simulation (a) and on a real patient (b) [21]

FAI treatment can also be done by surgical hip dislocation, an open surgery. In this technique, the doctor will perform an incision close to the greater trochanter figure 2.13a and safely dislocate the femoral head from the socket 2.13b so that the entire hip joint is exposed. Since the joint is totally dislocated it is possible to better

repair labrum tear and bony abnormalities of the femoral head, when comparing to the hip arthroscopy. Usually, this procedure takes between two and three hours to perform and is more invasive than the Hip Arthroscopy, so should only be considered when the extensiveness of the damage is considerable [20].

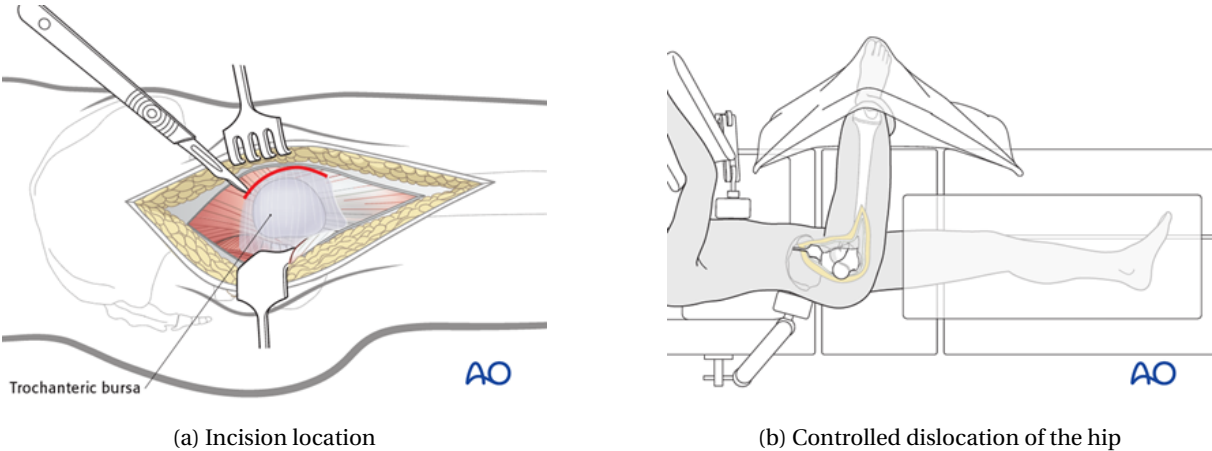


Figure 2.13: Surgical hip dislocation procedure [22]

Another possible type of surgery is a mini-direct anterior approach. This technique combines both previous ones but with a smaller incision when compared to traditional hip dislocation. In this procedure an incision is made in order to insert the arthroscope similarly to the Hip Arthroscopy surgery. Then, a 7-8 cm curved incision is made towards the femoral head [23]. The hip is slightly extend so that the surgeon can treat the labrum, the acetabular rim and all other internal damage in the hip bones by using a burr drill. Since this technique provides better visualization of the area to be observed it decreases the risk of damaging nerves and femoral cartilage when compared to Hip Arthroscopy [20].

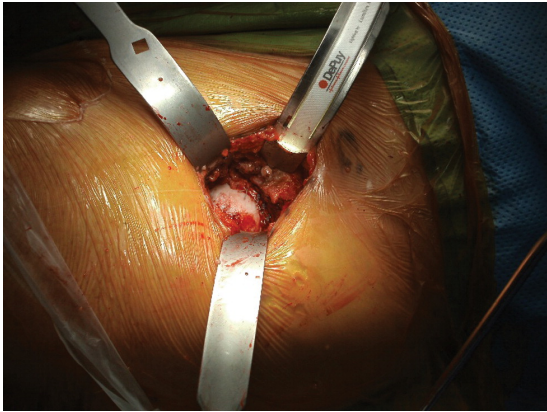


Figure 2.14: Mini-direct anterior approach [24]

2.4 State of the Art

Technological development had a major impact in the medical world with the implementation of new technology, such as digital imaging systems with higher quality when compared to traditional radiography. Further advances in technology and the desire to create safer protocols for surgeries were major influences to the state of the art of preoperative planning. New methodologies started to be part of everyday surgeries around the globe, naturally influencing orthopaedic practice as well. Today, it is possible to construct 3D digital models from imaging systems to better evaluate the case at hands as well as serving as a guide during surgery. Software has slipped into medical briefings and operating rooms and has proven to be a reliable teammate to increase the likelihood of success of surgeries. In this section, the state of the art of medical imaging software and preoperative planning tools will be presented.

2.4.1 3D Slicer

3D Slicer is an open source software platform for analysis and visualization of medical images and for research in image therapy. Born out of the necessity to find a more advanced quantitative imaging tool to better the personalization of cancer therapy, 3D Slicer was used to help building the Quantitative Imaging Network (QIN) by the American Nation Cancer Institute [25].

Fundamentally, 3D Slicer serves as a radiology workstation that provides not only versatile visualizations but also advanced functionality such as automated segmentation and registration for a variety of applications. The software supports multimodality imaging e.g. MRI, CT, ultrasounds (US), microscopy. The major advantage of 3DSlicer when compared to other radiology workstations is being open source and not tied to specific hardware. It is also available on multiple operating systems such as Linux, MacOSX and Windows, being easy to run on most computers.

The development of 3D Slicer is ensured by users, developers, contributors and commercial partners. This is a great advantage for researchers since it is a fully open source software and, thus, can be extended and distributed. A researcher can also create new functionalities through the creation of a 3D Slicer extension that is easily implemented.

One of 3D Slicer's drawbacks is the fact that it is not approved for clinical use. After being first introduced in 2012 [25] it is still only intended for research purposes so, using it in a hospital setting, such as an operating room, is not possible.

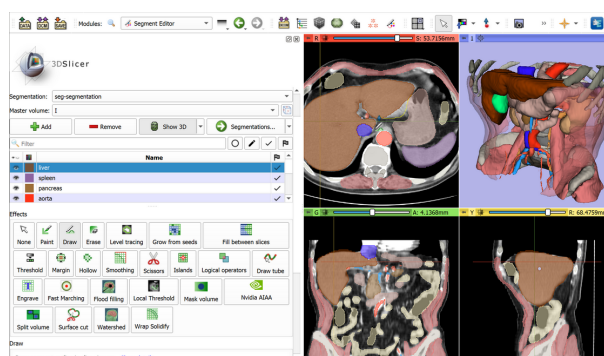


Figure 2.15: 3DSlicer layout [26]

2.4.2 OsiriX

OsiriX is a medical imaging software designed for navigation and visualization of multimodality and multidimensional images. The OsiriX project started as an open source software for Macintosh platforms specialized on DICOM images produced by medical equipment such as CT, MRI, Positron Emission Tomography (PET), etc [27].

Multidimensionality in OsiriX refers to the capacity of creating and visualizing 2D, 3D, 4D and 5D studies. The most important feature in OsiriX is its 3D Viewer that offers state of the art post-processing techniques, such as Multiplanar Reconstruction or 3D rendering. The 4D models are 3D series with temporal dimension such as Cardiac-CT and 5D models are similar with extra functional dimensions present in studies like Cardiac-PET-CT.

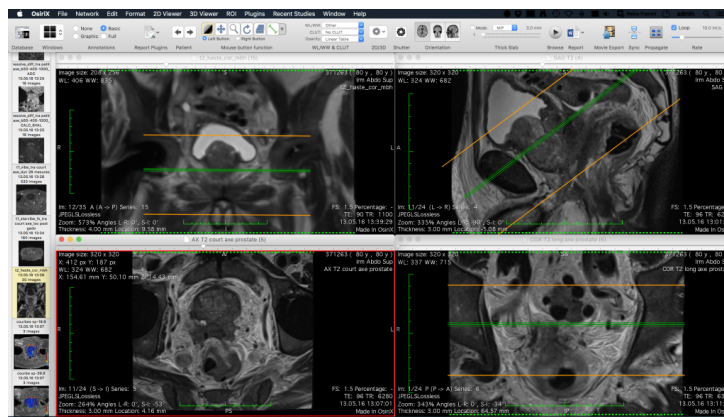


Figure 2.16: OsiriX layout [28]

Nowadays, OsiriX is property of Pixmeo SARL, a swiss company founded by the creators of OsiriX and is now a ISO 13485 certified medical device manufacturer, allowing OsiriX to be used in clinical setting. Then, OsiriX is no longer an open source software which means that its availability for medical or research purposes is limited. Another drawback of OsiriX as a DICOM viewer is its exclusivity to macOS running devices being unavailable for Windows or Linux operating systems.

2.4.3 DYONICS PLAN

In 2014, the medical equipment company Smith & Nephew launched DYONICS PLAN Hip Impingement Planning System, a software designed for FAI identification and correction planning. This software can create 3D rendering of a patient's hip based on CT scans and identifies the areas and the degree of hip impingement. It can also create range of motion simulations that allow better visualization to the user. DYONICS PLAN offers the unique ability of being able to identify areas of conflict and develop a resection planning based on stored coverage data from 400 asymptomatic hips. Furthermore, it allows the user to enter their clinical judgement, giving a doctor control of the planning [29]. The software also delivers an output report with the resection planning and customized different measures and views from the planning

Some of the disadvantages of DYONICS PLAN are the fact that it is not an open source software, it is limited to only the Hip Impingement pathology and its models are created in a proprietary format that is only used

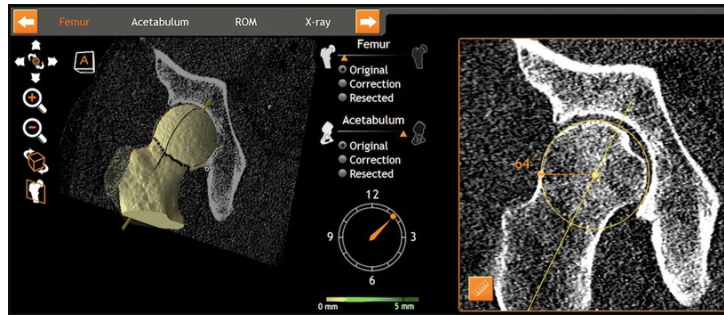


Figure 2.17: DYONICS PLAN layout [30]

by PLAN. This means that, as a research tool it is not the best software available. Another point to note is that the software is no longer available at Smith & Nephew's website and no information about its current state was found.

2.5 Surgical Robotics Lab and previous work

This thesis comes as a follow-up to already published research about preoperative planning for cam type FAI, more specifically, to two master's degree thesis developed in the Surgical Robotics Lab (SRL) of idMEC (Instituto de Engenharia Mecânica - IST).

The first work, [2], examined a way of applying a statistical shape model of a set of healthy femora to compare it with a pathological femur. In this thesis though, the used data to obtain the statistical femur was a 2D coronal section of each femur (figure 2.18). This will be the baseline of present thesis but it is necessary to understand that any human bone, such as the femur in these works, has a high degree of complexity, thus it is necessary to perform a 3D analysis that better depicts the characteristics of a real bone.

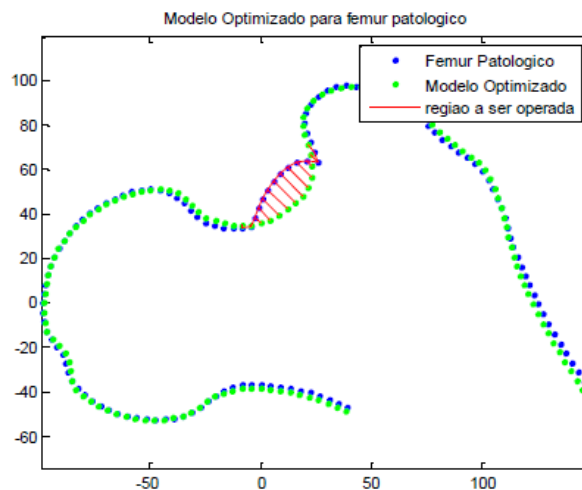


Figure 2.18: Difference between pathological femur (blue) and statistical model (green) shown in red [2]

In [31], the main goal was to develop a preoperative planning tool to be part of a robot-assisted conservative hip surgery. In said tool, it would be possible to outline, in each pathological femur MRI slice, what would be the normal geometry of a healthy femoral head and neck. From these outlines a surface would be

generated in order to serve as the safe zone to which the robot arm would be constrained to work. From this work, it is possible to understand the limitations of Image Segmentation when it comes to medical images, such as the ones used in this paper. It is important to state that the pathological femur medical images used in this thesis are the same as in [31] but the segmentation process was implemented independently, so minimal variation on the pathological femur may occur.

Chapter 3

Curan Software Development

Curan is an ongoing project being developed at the SRL that aims to create an open source medical visualizer based on several state of the art libraries. The project is completely written in C++ in order to be compatible with most operative systems. The goal is to build an application that supports visualization of medical images and preoperative planning, communication with external peripherals, such as robotic systems for Image Guided Therapy (IGT), and the flexibility to easily integrate new tools by developers.

3.1 Build and API

The tool used to build the project library is CMake. CMake is an open source, cross platform family of tools designed to build, test and package software [32]. To build a library, it is used along native build environments (Microsoft Visual Studio 2019 in this case). A configuration file is necessary to create standard build files, generally called *CMakeLists.txt* written in CMake language. In this file both libraries and executables can be added. It is in this file that all Curan's external libraries will be added.

Some important libraries are:

- Asio - This library is used for network and low-level input/output programming [33]. In Curan, it is used to establish TCP/UDP protocol communication with external applications;
- GeometricTools - Used for computation in the fields of mathematics, geometry, graphics and image analysis [34]. Some important mathematical operations depend on this library, such as convex hull estimation and smallest bounding box;
- GLFW - One of the most used API for creating and managing windows, contexts and surfaces [35];
- ITK - Insight Toolkit (ITK) is a library that provides extensive tools for image analysis. Capable of reading and writing DICOM files (one of the most used file types for medical images), it is possible to perform segmentation and/or registration in multidimensional images [36];
- nlohmann_json - Creates JSON files used to record information for different sessions of the application;

- OpenIGTLink - Provides a standardized protocol for the communication between computers and medical devices for image guided therapy. Some examples of this library use are robot-assisted interventions, with robotic devices and surgical planning software, and intraoperative image guidance using real-time MRI and medical image visualization software [37];
- sigslot - Generates signals between widgets and other library components;
- Spdlog - One of the fastest C++ loggers, this library allows multithreaded information to be output to the command line;
- stb - Used for loading images.

Currently, there are two external dependencies that are not directly injected into the source code, Vulkan and Skia. Vulkan is a new generation low-level graphics API best known for its high performance and used as a state of the art API for video game development. A graphics API works as a "middleman" between the software and hardware, providing access to the graphics processing units (GPUs) [38]. When compared to other APIs, such as the predecessor OpenGL, Vulkan is a considerably lower-level API, reason why it is such a powerful tool for 3D rendering at a higher performance. Skia is an open source 2D graphics library developed by Google and it is used as the graphics engine for popular products such as Google Chrome, Mozilla Firefox or Android [39]. In Curan, it will be used to create and display 2D entities on the screen.

Curan API is divided into four logical groups based on the specialization of each group inside the application. All classes are contained in the `curan` namespace.

- Image Processing - This group focus on the medical imaging management, such as storage in RAM memory and algorithms related to segmentation, registration, volumetric reconstruction, etc. The code is contained inside `curan::image` namespace;
- Display - The code in this group takes care of widget functionality, such as buttons, and information to be displayed in 2D to the user. It is contained in the `curan::display` namespace;
- Communication - This group is in charge of the communication between Curan and external peripherals, such as surgical robots or ultrasound, by using the TCP/IP protocol. The code is contained in `curan::communication`;
- User Interface - This portion controls the behaviour between the Display group and window management and logic. The namespace `curan::ui` contains all the windows that can be shown to the user by the application.

3.2 FAI Preoperative Planning

Given the flexibility of integration and the tools presented earlier, the goal is to start the implementation of a window in the Curan app dedicated to the preoperative planning of the cam type femoroacetabular impingement pathology presented in section 2.3.3. For this, alongside the basic functionalities of both 3DSlicer and Osirix in image processing, such as segmentation and registration, that are necessary for any state of the art medical imaging visualizer, it is intended to implement an algorithm that would help identifying where the bony deformation occurs and the shape of the region to be surgically removed. For this, it is intended to build a 3D Statistical Shape Model (SSM) of the top part of the femur. A Statistical Shape Model is a model capable of characterizing the shape of an object based on the mean shape of a training set and a set of modes of variation of said shape [40]. This approach is often used on objects that have a recurrent varying shape as it is the case of human bones.

The Statistical Shape Model of the human femur to be used in Curan was implemented and developed in MATLAB® R2021a due to the powerful data analysis and visualization tools it inherently has. The final data resulting from SSM implementation can easily be exported to be used in Curan. Even though the segmentation tool was implemented in C++ and explained in subsection 3.2.2, the segmentation of the pathological femur, used for the results of this thesis, was obtained in MATLAB® due to trustworthy of results when compared to the emerging Curan software that is still susceptible to changes.

3.2.1 Medical Imaging Visualizer Implementation

Before diving in the implementation of the visualizer it is necessary to understand the foundation of Curan and the logic behind how the application works.

The main source file, `app.cpp`, is where function `start_up()` is defined. In this function, before constructing the window, where the medical imaging and options will be presented, there are five initialization steps needed:

1. Create a pointer to the thread that is intended to be used. This is important since Curan is built as a multithreading program that is able to execute different actions at the same time (in different threads) but communicating with each other (concurrently);
2. Load the resources necessary for the layout of the Curan app, mainly the button icons;
3. Load the desired clinical study into memory under the class `curan::display::StudyManager`. This is currently done by using the file path of the file with the medical imaging;
4. The method `initialize_context()` from the `curan::display::Context` class will create the Vulkan context instance necessary to communicate with the GPU;
5. Finally, it is necessary to define `curan::display::DisplayParams` that will initialize the parameters of widgets and overall aspect of the window.

Now, it is necessary to call the class that will manage the information to be displayed to the window, `curan::ui::WindowManager`. As the name suggests, this class will manage what window will be called

and the necessary methods to perform this. Firstly, the manager will set the previously created Vulkan context with `set_context()` method. Then, `launch_faiviewer_visualizer()` method will launch the FAI planning window. At the same time, the `process_os_events` method is called. This important method will monitorize all pending events in the created window. It will process the events in the event queue, such as input callbacks like mouse clicks.

The `launch_faiviewer_visualizer()` method will create an instance of the `FAIViewer` class that holds the methods necessary for the FAI planning window. Firstly, the `init()` method is called to setup the initial state of the FAI window, mainly the disposition of widgets, such as buttons, size and initialize the Vulkan context for this window. It will also set the type of input that can be received through `connect_handler()` method, that in this case will be the cursor position, mouse buttons and scroll wheel. Then, the `run()` method is called. This method will manage the behaviour of actions in the window while the app is running. It is in charge of drawing images to the screen and decide on actions based on user input signals. The FAI window is shown in figure 3.1.

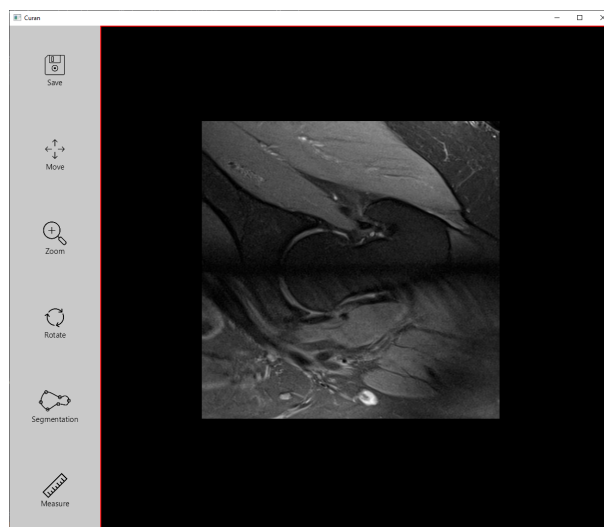


Figure 3.1: FAI Visualizer in Curan

From figure 3.1 it is possible to observe the existence of six buttons in this window. From top to bottom:

- Save - Saves the changes made to the current image and stores this changes in memory;
- Move - Allows to freely move the image in the designated image area (area with black background);
- Zoom - Allows to zoom in and out on a specific area of the image;
- Rotate - Gives the option to freely rotate the image around its center point;
- Segmentation - This button opens the segmentation options box, further explained in subsection 3.2.2;
- Measure - Through a series of mathematical relations, this button allows to measure the distance between two points selected in the image.

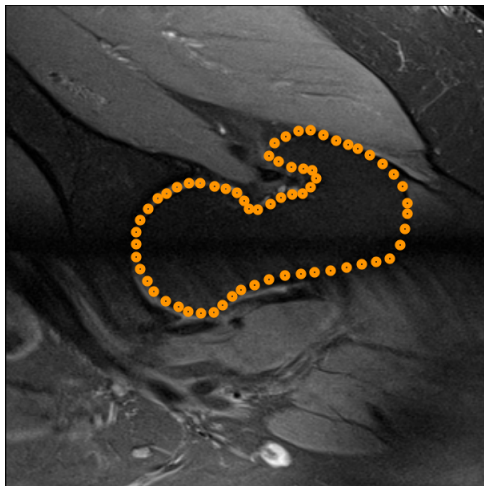
These buttons allow for a similar experience one has with medical imaging software. Options like moving, zooming or rotating an image are important when analysing a clinical study. Bear in mind that these are the available button as of the time of writing this thesis and in the future are subject to changes.

3.2.2 Manual Segmentation Options Box

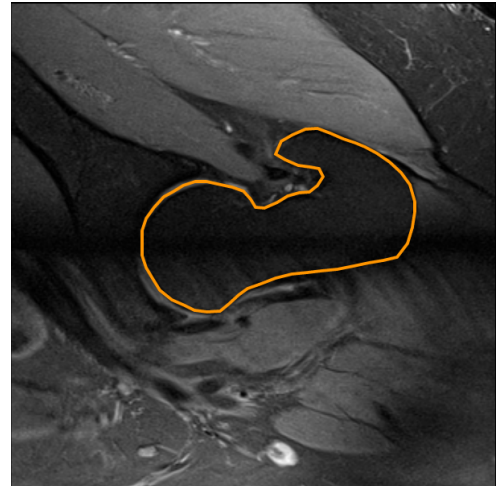
The segmentation of a medical image is known to be a difficult task. The biological variation, the quality of the image acquisition and different pathology all influence the task of identifying the object of interest. While Image Segmentation is a widely studied field on itself, and automated methods have been applied to medical studies in platforms, such as 3DSlicer or OsiriX, these options are still not implemented in Curan. For the time being, it was implemented a way of manually drawing the contour of a region of interest (ROI).

To start the manual segmentation of a medical image, the user should press the *Segmentation* button in the FAI window. This will open a small menu-like box on the top left of the image area. This box will feature five different buttons each with a different function.

- *Adjust Contour* allows to edit a previously drawn contour by dragging the existent vertices of said contour;
- *Hide Contour* gives the option to hide/show an existing contour;
- *Add Points* allows the user to start drawing a contour. After pressing this button the user will start drawing points on the image (figure 3.2a);
- *Submit Contour* will save the current contour as a path. This will save the coordinates of the vertices drawn by the user and draw lines connecting all points (figure 3.2b). It is to note that these coordinates are the (i, j) pixel coordinates of the image with size $i \times j$ pixels;
- *Erase* will allow the user to select a contour to eliminate. This will delete the contour from the screen and its coordinates from memory.



(a) Point placement



(b) Closed contour

Figure 3.2: Steps of drawing a contour in Curan

In order to achieve the real coordinates of the human femur it is necessary to apply the mathematics presented in section 2.2 using the (i, j) coordinates drawn on the image. After repeating the procedure for all radial MRI slices, it is possible to construct the 3D model of the patient's femur. For now, one of the biggest drawbacks is that the number of points defining a slice contour is defined by the number of points the user

draws on the image. In the future, it is intended to implement a way of creating a spline curve, based on the points drawn by the user, with equally distanced points.

Chapter 4

Statistical Shape Model

Statistical Shape Model (SSM) is a technique of analyzing an object by extracting the mean shape of said object alongside a number of modes of variation from a training sample set that is specially useful in medical applications due to natural variability between all humans [40]. The main goal of the chapter is to obtain an SSM of the proximal femur from a set of 33 human femora and identify cam type FAI detection with the model.

4.1 Algorithm

The SSM algorithm starts with obtaining the point coordinates, landmarks, of an object, in this case the proximal femur. The array of k points defining a femur is stored in vector \mathbf{x} [40].

$$\mathbf{x} = (x_1, y_1, z_1, \dots, x_k, y_k, z_k)^T \quad (4.1)$$

With the landmarks a Point Distribution Model (PDM) of each femur in the training set is obtained [41]. One of the necessary requirements to build a PDM is that the landmarks on the different training samples need to be located at corresponding positions, this is, a landmark point needs to be located in the same relative position in all samples. In figure 4.1 it is possible to observe the assigned position to each of the points defining the shape of a resistor. In order to build a PDM for this resistor it is important to maintain the points in around the same position for other resistors in the training sample e.g. points 3, 5, 10, 12, 10, 21, 26 and 28 should be placed in the "vertices" of all resistor bodies in the training set. Another requirement is that all models in the training set have the same number of points in order to obtain a point to point correspondence for the mean shape model.

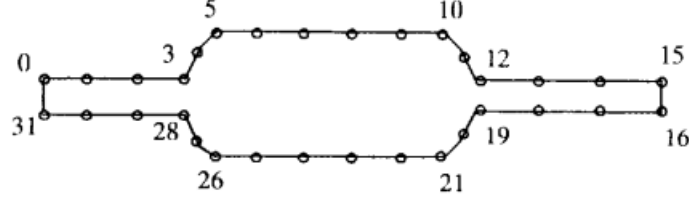


Figure 4.1: Thirty-two point model of the boundary of a resistor from [41]

With all the points aligned in different training samples it is then possible to compute the mean shape of the object, $\bar{\mathbf{x}}$, in study.

$$\bar{\mathbf{x}} = \frac{1}{N} \sum_{i=1}^N \mathbf{x}_i \quad (4.2)$$

the covariance matrix, S , is then calculated with all shapes in the training set, \mathbf{x}_i , and the mean shape, $\bar{\mathbf{x}}$ using equation 4.3

$$S = \frac{1}{N} \sum_{i=1}^N (\mathbf{x}_i - \bar{\mathbf{x}})(\mathbf{x}_i - \bar{\mathbf{x}})^T \quad (4.3)$$

From the covariance matrix it is then possible to use eigendecomposition in order to find the eigenvalues, λ_k , and correspondent eigenvectors, \mathbf{p}_k , that describe the most significant modes of variation as explained in [41]. From [40], a more numerically stable method to discover \mathbf{p}_k and λ_k is by performing Singular Value Decomposition (SVD) to the aligned landmark matrix L , where each column is the difference between each femur in the database, and the mean shape, $(\mathbf{x}_i - \bar{\mathbf{x}})$, in order to obtain U , the matrix of the eigenvectors of L and D , the diagonal matrix with respective eigenvalues, λ_i . SVD is an algorithm that provides a numerically stable matrix decomposition that can later be used for different types of analysis [42]

$$L = \begin{bmatrix} \vdots & \vdots & \vdots \\ (\mathbf{x}_1 - \bar{\mathbf{x}}) & \dots & (\mathbf{x}_i - \bar{\mathbf{x}}) \\ \vdots & \vdots & \vdots \end{bmatrix} \quad (4.4)$$

From here, any allowable shape will be the result of a linear combination of the t relevant eigenvectors and the mean shape, resulting in

$$\mathbf{x} = \bar{\mathbf{x}} + \mathbf{P}\mathbf{b} \quad (4.5)$$

where $\mathbf{P} = (\mathbf{p}_1, \mathbf{p}_2, \dots, \mathbf{p}_t)$ is the matrix of the first t relevant eigenvectors and $\mathbf{b} = (b_1, b_2, \dots, b_t)$ is a vector of weights. t is usually chosen such that the cumulative variance $\sum_{i=1}^t \lambda_i$ reaches 90-98% of the total variance $\sum_{i=1}^k \lambda_i$. The weights b_i will be the parameters that will affect the shape of the Statistical Shape Model so it is important to define limits in between which they can vary. The variance of these weights over the training set is given by its correspondent eigenvalue, λ_i , so a suitable limit for b_i is within three standard deviations of the variance.

$$-3\sqrt{\lambda_i} \leq b_i \leq 3\sqrt{\lambda_i} \quad (4.6)$$

This method to obtain an SSM is considered to be better for shape analysis models or diagnostic purposes. It is similar to PCA (later approached in this thesis) but instead of delivering global modes that influence all variables in the model, this method results in modes that will influence the model in a more isolated way, allowing for a more intuitive interpretation. So, each mode will affect local clusters of data or specific regions in the model [40].

4.2 Training Set

To derive a statistical shape model of a human femur, as explained in section 4.1, a data set is needed. This data set should contain the coordinates of points located on the surface of a human femur in order to derive 4.1. The Osteoporotic Virtual Physiological Human project (VPHOP) is a research project funded by the European Union and other institutions that aims to accurately predict a patient-specific risk of fracture for osteoporotic patients. During the course of this investigation project an open-source database of human proximal femora was published by the Eindhoven University of Technology [43]. This database featured a total of 33 human cadaver proximal femora with ages varying from 61 to 95 years old donors with a mean of 77.8 ± 10 years. 17 out of the 33 femora were from female donors and the remaining 16 from male ones. This database featured a set of 3 files for each femur, a finite element model input file in ANSYS format (*.cdb files), a tabular file containing microstructural parameters measured at the location of each element (*.dat files) and the reconstructed microCT images of the sample (*.aim files).

In order to acquire data, it is necessary to get the nodes of each of the femora in the database. To do so, ANSYS Software was used. By uploading the *.cdb files to ANSYS Workbench as an External Model and analysing them in ANSYS Mechanical (figure 4.2), it was possible to export the surface mesh information from each femur data to a *.txt file.

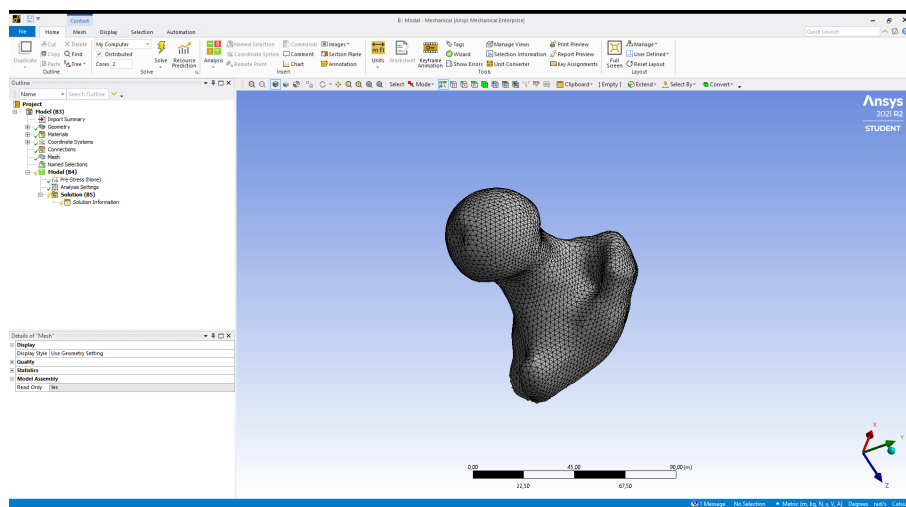


Figure 4.2: Femur FEM01098 from VPHOP database in ANSYS Mechanical

Additionally, viewing the 3D model femur in ANSYS Mechanical allows for correctly labeling whether it is

the right or left femur, based on the location of the greater and lesser trochanters (figure 2.1). This will later become important since a reflection transformation will have to be applied to the femora of one side in order to achieve the SSM of the femur.

With this process, 33 files were generated, each containing 3 columns of information relative to each node in the finite element model. The columns have the (x, y, z) [mm] coordinates, necessary to building vector 4.1. The *.txt files were then uploaded into MATLAB® R2021a and the information on the number of nodes and side is shown in table A.1.

4.3 Point Registration

Point Registration is the process of aligning a set of data, in this case the coordinates of the femora database, in order that each femur holds its shape but all femora are placed as close as possible [44] so that it is later possible to compute the Statistical Shape Model as explained in 4.1. The process of overlaying different objects can be divided into two main parts, coarse registration and fine matching [45]. For the first, Principal Component Analysis (PCA) will be used so that the principal components of each femur are aligned. For the second, the Iterative Closest Point (ICP) algorithm will provide a fine point by point match so that the final Statistical Shape Model better reflects the characteristics of each femur.

In figure 5.2 it is possible to observe three femora with different spatial orientation. The remaining femora in the database have one of these three orientations but the majority of the femora have a similar position to femur 4.3c.

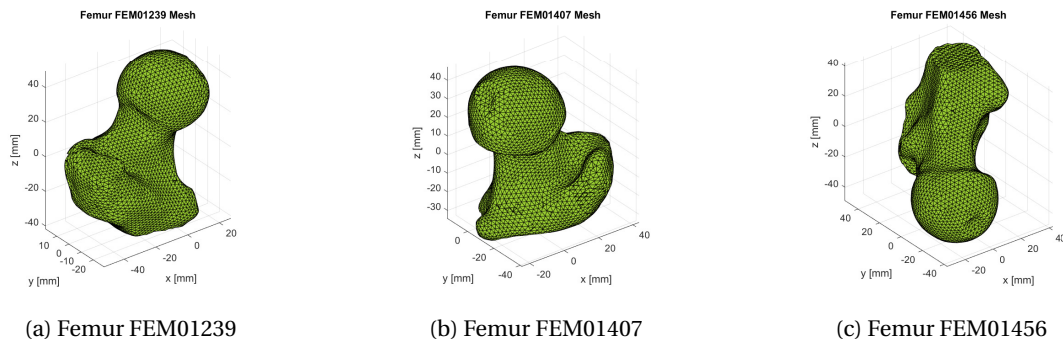


Figure 4.3: Meshes from three different femora in the database

Before advancing into the registration algorithms, it is important to remember that 14 out of the 33 femora in the database are left femora, so in order to compare left and right models a reflection by the xOy plane will be applied to the left ones by multiplying the z coordinate by (-1) (figure 4.4).

4.3.1 Principal Component Analysis

One of the most important algorithms for Dimensionality Reduction and useful linear transformation techniques is the Principal Component Analysis (PCA). It is based on Singular Value Decomposition (SVD).

Principal Component Analysis (PCA) is one of the most used algorithms in Data Science. This algorithm creates a hierarchical coordinate system that represents high-dimensional data depending on the variance

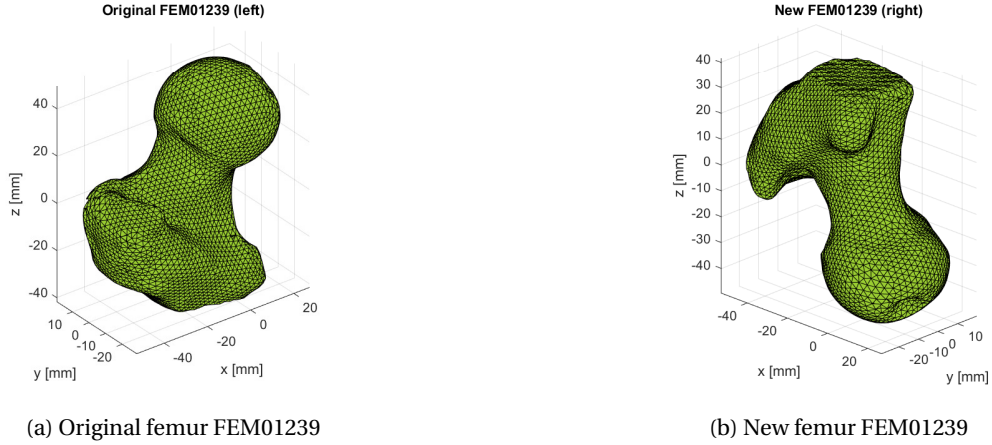


Figure 4.4: Original left femur FEM01239 and after reflection by the xOy plane

of the data set. This coordinate system will be built based on the way that the data is spread, in this case, the number of points in a certain region will affect the direction of this coordinate system.

The first step of PCA is to center the data. This means calculating the mean point, $m_k = (x_m, y_m, z_m)$, in each femur, k , and subtracting it to all points, i , in the femur's matrix.

$$m_k = \frac{1}{N} \sum_{i=1}^N (\mathbf{x}_i, \mathbf{y}_i, \mathbf{z}_i) \quad (4.7)$$

The centered femur is given by subtracting the mean m_k from all points in the original femur matrix X ,

$$\mathbf{x} = X - m_k \quad (4.8)$$

After the data is centered, the covariance matrix is obtained by,

$$C = \mathbf{x}^T \mathbf{x} \quad (4.9)$$

By computing the eigendecomposition on C it is possible to obtain the eigenvectors of C ,

$$C = V D V^T \quad (4.10)$$

where V is the matrix of the eigenvectors, the principal components of C , and D the diagonal matrix with the eigenvalues corresponding to the matrix V columns. In figure 4.5, the principal components of two femora are shown. It is to note that the blue vector is the first principal component (first column of V), the red vector the second principal component (second column of V) and the orange vector is the third principal component (third column of V). In figure 4.5, it is possible to observe the principal components from femora FEM01239 and FEM01456.

Finally, since the eigendecomposition was applied on the covariance matrix of the centered centered data, the columns in V will be orthornormal basis vectors of said femur (in figure 4.5 the eigenvectors were multiplied by 70 for visualization purposes). The last step is to multiply the transpose of V , V^T , by the data in order to align all principal components from the set with the frame of reference.

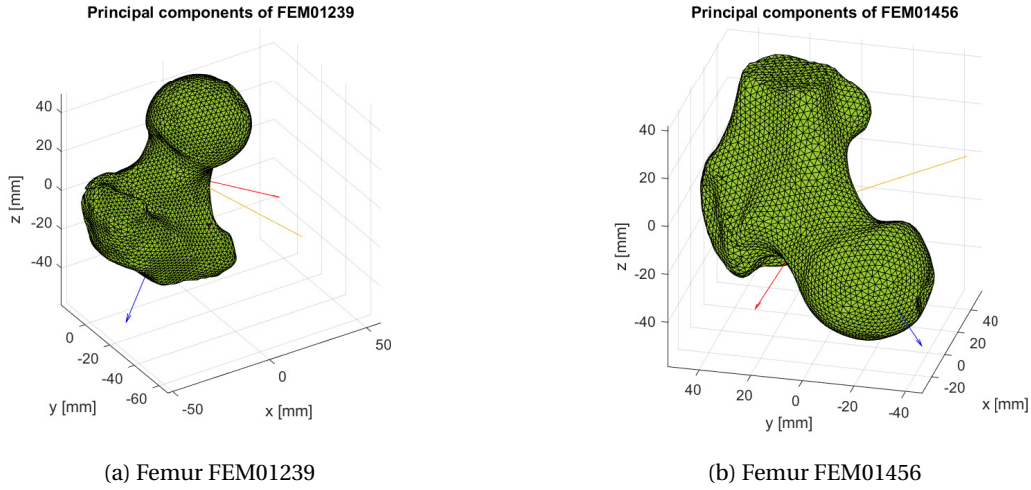


Figure 4.5: Principal components plotted on respective femur

4.3.2 Data Refinement

Recalling subsection 4.3.1, the orthonormal basis of each femur is heavily influenced by the number of points in a specific region, reason why the principal component (blue vector in image 4.5) is the one going through the head, neck and part of the femur shaft.

During inspection of the three dimensional plots of the femora, it was found that some femora in the database featured more volume of the shaft than others. This means that the principal components analysis will be influenced by the larger number of points in the shaft of these femora. In figure 4.6, it is possible to observe that femur FEM01407 has a smaller volume of the shaft when compared to FEM01462.

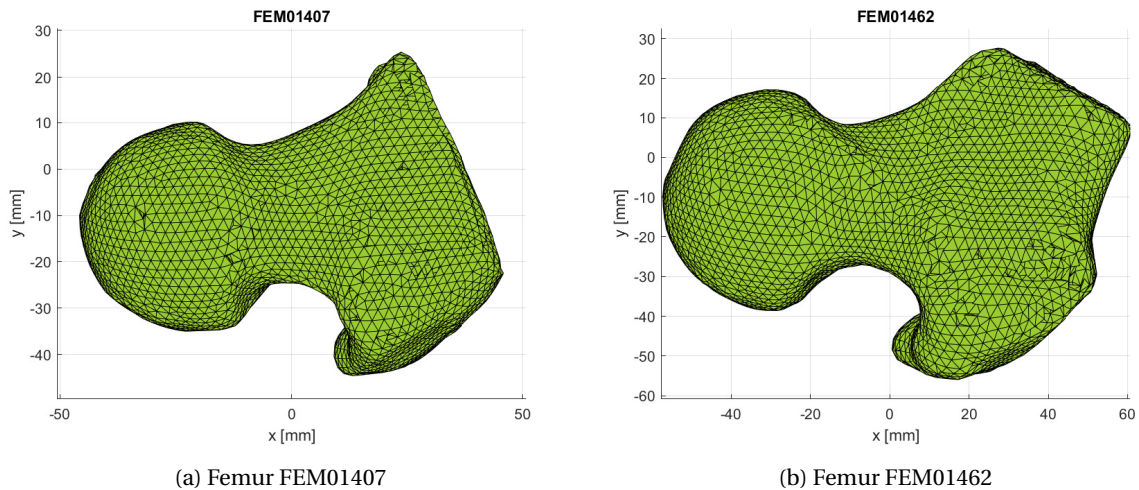


Figure 4.6: Anterior view of two femora with different shaft angle

Based on the angle of the shaft on femur FEM01407, a plane with equation

$$y = -1.5x + 62 \wedge z \in \mathbb{R} \quad (4.11)$$

was considered, and all points above it were eliminated from all femora (figure 5.4a. The updated number of

nodes for each femur is presented in table A.1. After this reduction in volume, a new PCA was computed and the coarse registration process is finished.

4.3.3 Iterative Closest Point

The Iterative Closest Point (ICP) algorithm is one of the most powerful methods to achieve a more precise point cloud matching. In this work it is used as a refinement technique of the registration performed by PCA so that minor adjustments, not detected by the principal components, can be made. ICP's goal is to find an optimal mapping by means of a rigid transformation, T , (translation and/or rotation) between the target point set, S , and the reference point set, M , that minimizes the least squares error of the distance between corresponding points [45].

Assuming that the coordinates of S are $\{S_i | S_i \in R^3, i = 1, 2, \dots, N_S\}$, the coordinates of M are $\{M_i | M_i \in R^3, i = 1, 2, \dots, N_M\}$ and, in the k -th iteration, the coordinates of M corresponding to the coordinates of point set S are now $M^k = \{M_i^k | M_i^k \in R^3, i = 1, 2, \dots, N_M\}$ [46]. The first step is to find the corresponding point $M_i^k \in M^k$ in set M such that

$$\|M_i^k - S_i^k\| = \mathbf{min} \quad (4.12)$$

Then, the rotation matrix, R^k , and the translation vector, T^k , are calculated so that

$$\sum_{i=1}^N \|R^k S_i^k + T^k - M_i^k\|^2 = \mathbf{min} \quad (4.13)$$

Then, the target set point is updated by multiplying the rotation matrix R^k and summing T^k ,

$$S^{k+1} = \{S_i^{k+1} | S_i^{k+1} = R^k S_i^k + T^k, S_i^k \in S\} \quad (4.14)$$

The distance d^{k+1} between the new point set S_i^{k+1} and M_i^k is calculated,

$$d^{k+1} = \sum_{i=1}^N \|S_i^{k+1} - M_i^k\|^2 \quad (4.15)$$

Given a threshold τ for the maximum distance between corresponding points in the final model, the algorithm is repeated until $d^{k+1} < \tau$. In MATLAB® R2021a, the ICP algorithm was implemented by using function `pcregistericp` from the Computer Vision Toolbox [47]. This function has the following inputs and outputs,

$$[\mathbf{tform}, \mathbf{movingReg}, \mathbf{rmse}] = \mathbf{pcregistericp}(\mathbf{movingPC}, \mathbf{fixedPC}) \quad (4.16)$$

where:

- `movingPC` is the point cloud of the target set point S ;
- `fixedPC` is the point cloud of the reference set point M ;

- `tform` is the rigid transformation, T , with the rotation matrix R^k and translation vector T^k ;
- `movingReg` is the transformed point cloud, result of the rigid transformation applied on the moving point cloud `movingPC`;
- `rmse` is the root mean square error that represents the Euclidean distance between the inlier aligned points.

In order to improve the performance of the ICP algorithm, every femur was normalized. In this way, changes in femur size due natural variance are neglected allowing to have a better point to point mapping.

4.4 Point Remapping and Mean Shape model

As explained in 4.1, to build the SSM of the proximal femur it is necessary to ensure that the points in matrix \mathbf{x} from 4.1 are in the same order. For example, the first point in all matrices \mathbf{x} should be placed in the same place in all femora, as well as the second and so on.

Due to different number of nodes characterizing each femur (min = 3172 nodes and max = 3548, see table A.1), and based on the femur chosen to be used as reference, it is important to downsample said femur so that it is possible to remap remaining femora to only having points corresponding to points in the reference. To do so, MATLAB®'s function `pcdownsample` was used.

$$\text{downsampledPC} = \text{pcdownsample}(\text{pointCloud}, \text{'gridAverage'}, \text{gridStep}); \quad (4.17)$$

where,

- `pointCloud` is the point cloud of the data set intended to downsample;
- `'gridAverage'` is one of the possible downsampling methods. In this method, points within the same box will be merged as a single point in the output. Other possible method is the `'random'` method that eliminates random points in the data set based on a percentage of points to keep;
- `gridStep` is the size of each 3D box;
- `downsampledPC` is the downsampled point cloud.

Due to a small range between the femora with the lowest and highest amount of nodes, it is expected that the reference femur will only be downsampled to a value slightly lower than the femur with the minimum point number (3172) so, the point density along the reference femur should not change much. Following the downsampling of the reference femur, it is now necessary to remap the points in every femur to the same order as the reference.

Let us take a k candidate femur as \mathbf{X}^k , with $k = 1, \dots, 32$, an $m \times 3$ matrix and the reference femur as \mathbf{X}^{ref} an $n \times 3$ where m and n represent the number of points in each matrix. The goal is to obtain the remapped matrix \mathbf{X}^k , a $n \times 3$ matrix where each row represents the closest point of \mathbf{X}^k to the same row point in \mathbf{X}^{ref} . The implemented algorithm steps are as follows:

1. For each k femur, compute Δ , an $m \times n$ matrix where,

$$\Delta_{ij} = \|\mathbf{X}_i^k - \mathbf{X}_j^{\text{ref}}\|^2 \quad (4.18)$$

with $i = 1, \dots, m$ and $j = 1, \dots, n$. Each entry Δ_{ij} represents the euclidean distance between each point i in the candidate femur and the point j in the reference femur.

2. Compute the minimum value of matrix Δ , δ_{ij} . This value represents that point \mathbf{X}_i^k is the closest to point $\mathbf{X}_j^{\text{ref}}$. This means that point \mathbf{X}_i^k will be remapped into \mathbf{X}_j^k .

$$\mathbf{X}_j'^k = \mathbf{X}_i^k \quad (4.19)$$

3. Since it is not intended to repeat or replace points in matrix \mathbf{X}'^k , after remapping a point \mathbf{X}_i^k to \mathbf{X}_j^k both row i and column j of Δ are replaced with infinity. After this, repeat from point (2) until femur k is totally remapped into \mathbf{X}'^k and repeat from point (1) to remapped a new femur.

After the remapping is finished and all \mathbf{X}'^k matrices are computed, the final step is to denormalize each femur with the normalizing factor used before and apply equation 4.2 to the obtained matrices and compute the set of points that describe the mean shape of the femur based on 33 femora from the database. The mean shape of the proximal femur can be observed in figure 4.7.

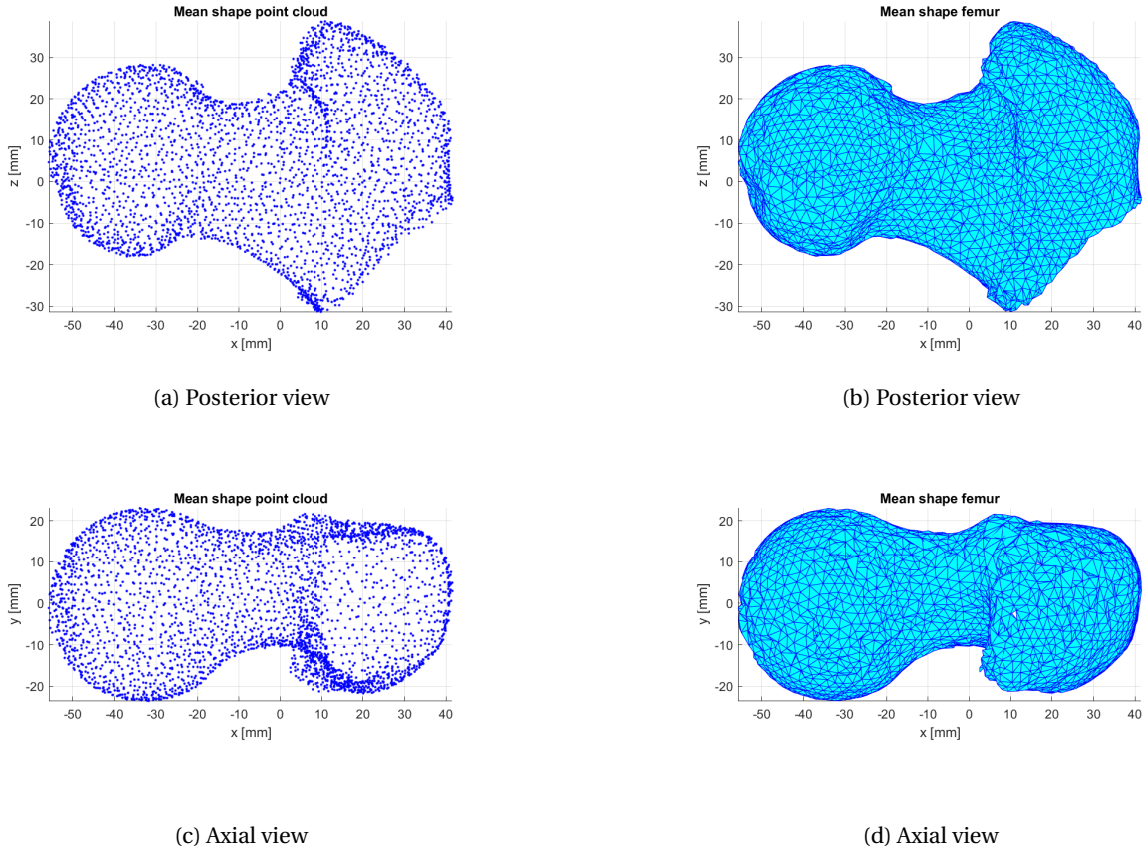


Figure 4.7: Point cloud and mesh of the femoral head and neck mean shape

4.5 Pathological Femur model

With the mean shape of the proximal femur, it is now necessary to obtain the model of the femur with the FAI pathology and compare the differences between models.

The images of the femur with FAI were obtained at Hospital da Luz using their standard procedure of collecting 24 radial slices that results in 48 images after post-processing [31]. A series of 48 DICOM files is generated with the necessary information mentioned in section 2.2.

Given the extensive analysis made in previous work [31], on possible state of the art segmentation tools, to isolate the femur from remaining soft tissue and bones, it has proven to be challenging to automatically isolate a region of interest, such as the femur. This is due the natural variance in shades in different osseous tissue (cortical and trabecular bone) and the crosstalk artifact, mentioned in section 2.2. So, to acquire the pathological femur model, the segmentation will be performed manually, as in [31].

Similarly to the point placement approach implemented in Curan (figure 3.2a) the same was done in MATLAB® R2021a with the implementation of function `ginput` that returns the (i, j) pixel coordinates of the points selected. Then, a spline was built based on these points with N equally distanced points in order to generate a higher number of points to better describe the geometry of the femur. In figure 4.8 the result for one on the images is shown, with the red points being the user selected points and the cyan are the equally distanced points across a spline. A contrast adjustment was applied to the original image to better the visibility of the femur's contour.

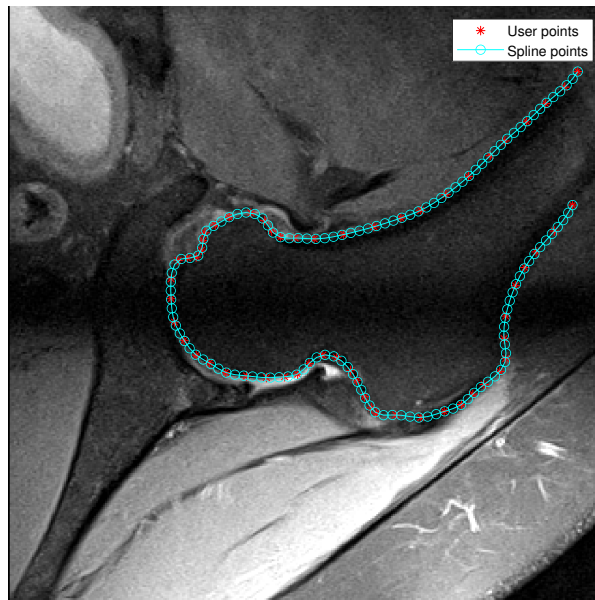


Figure 4.8: Manual segmentation of a radial slice in MATLAB for $N = 130$

After defining the region of interest, it is necessary to transform the (i, j) coordinates into the real (x, y, z) coordinates with the help of equation 2.1, resulting a $N \times 3$ matrix. After repeating the process for all 24 MRI

images, all matrices were merged to obtain the full 3D representation of the pathological femur (figure 4.9). Therefore, the resulting data was treated similarly to what was done before to obtain the mean shape of the healthy femur. Operations applied before, such as centering the data (equation 4.8) and reducing the volume of the shaft (from subsection 4.3.2) were applied to the pathological femur as well in order to prepare the data set for registration.

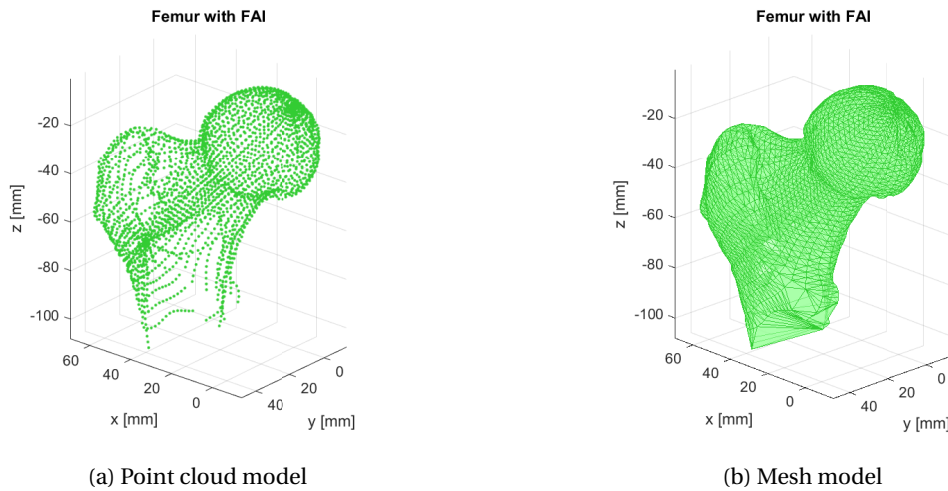


Figure 4.9: Pathological femur representation in space

4.6 FAI Identification

To proceed with the comparison of the pathological femur with the statistical one to identify the FAI deformation, it is necessary to, once again, proceed to register both femora. The general idea for the registration is the same used in section 4.3. Firstly, from the resulting models in figure 4.9, one can observe that the pathological femur is from the left side of the human body so the y column of the mean shape femur data will be multiplied by (-1) so that the statistical femur is from the left side. Secondly, since the pathological femur point cloud was achieved by manually segmenting each radial MRI slice, it is expected that the areas near the axis of revolution of the MRI machine, such as the femoral head and near the lesser trochanter, will have a higher point density. This can become a problem since such density can affect the PCA registration and subsequent remapping algorithms so, to correct this, the downsampling function 4.17 will be used again with the goal of averaging the number of points per grid box along the surface of the femur.

Once again, the registration will be separated in two parts. For the coarse matching, the principal components of both femurs will be acquired and the femora will be translated and rotated so that all principal components are aligned (figure 5.13). Then, the fine matching will be completed by using the `pcregistericp` function by taking the mean shape femur as reference and applying a rotation and translation to the pathological femur.

After registration is concluded, it is intended to understand the anatomical differences between the pathological and the statistical femur. The challenge now is to distinguish between natural variance in the pathological femur and the FAI caused anomalies. Firstly, it is once again applied the algorithm presented in 4.4

in order to have the closest points mapped to the same position in both femora matrices and to eliminate points that will not be important for the comparison. With the remapping the number of points in the mean shape femur will be reduced to the same as the pathological femur and the remaining points will be better aligned with the ones from the pathological model.

From 2.3.3, it is known, *a priori*, that cam type FAI deformation occurs on the anterior/superior aspect of the femoral head-neck junction. Without medical feedback, important since every human's anatomy is different, and only recurring to literature, this area is assumed to be where the pathology will manifest. In order to mathematically define this/these region/s a euclidean based distance clustering algorithm was used.

1. Calculate the euclidean distance between corresponding points from both models.
2. Compute the mean of the distances and reject outliers distancing more than three standard deviations from the mean value.
3. Recalculate the euclidean distance between the remaining points and select the ones from the FAI model that have a distance to the mean model greater than the mean.
4. Select the points placed around the anterior/superior region of the pathological femur and construct clusters using MATLAB® function `dbSCAN` that given a list of point coordinates builds clusters based on the density of points on a certain radius circle.
5. In order to recreate the three dimensional region that will have to be removed, it is important to also select the region were the pathological femur and the mean femur intersect, thus points in this region have a distance close to zero.

Chapter 5

Results and Discussion

5.1 Mean Shape femur

Acquiring the mean shape of the human femur is the first task into building a statistical shape model so, in order to do that, registration and remapping of the training set was performed as explained in 4.

One of the first things to consider before diving into the analysis of results is the quality of the database. As explained in section 4.2, the database features femora models collected from human cadaver proximal femora with ages varying from 61 to 95 years old donors with a mean of 77.8 ± 10 years old for a osteoporosis research project. Firstly, this age range is not representative of the population that can develop cam type FAI since this is a condition that heavily affects young people. Secondly, it is not know if these femora were from osteoporotic patients or healthy ones and, while osteoporosis only changes the microarchitecture of the bone [48], it is also unknown if the donors suffered from other medical conditions that could affect the geometry of the bone. The femora were compared with literature and were taken as healthy but feedback from a health professional, such as an orthopedist, would be favourable into validating the database. As explained in 4.3, the registration was divided into two parts, coarse and fine matching each resorting to PCA and ICP methods respectively.

5.1.1 PCA registration

While analysing the set of femora from the database it was perceived that the spatial orientation of each femur was different but could be divided into one of three groups, each represented by femora FEM01239, FEM01246 and FEM01456, each with a different color in figure 5.1. After the PCA algorithm was applied all femora were alligned by the coronal plane but a problem emerged where two of the the principal components of some femora did not have the same order or orientation, leading to registration errors like the one in figure 5.1b. To guarantee that the final result of the PCA method is the correct one, and since all femora were alligned by the coronal plane, the femora having a poor registration were multiplied by a rotation matrix, such as

$$R_x = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos t & -\sin t \\ 0 & \sin t & \cos t \end{bmatrix}, R_y = \begin{bmatrix} \cos t & 0 & \sin t \\ 0 & 1 & 0 \\ -\sin t & 0 & \cos t \end{bmatrix}, R_z = \begin{bmatrix} \cos t & -\sin t & 0 \\ \sin t & \cos t & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad (5.1)$$

with R_x rotating around the x axis, R_y around the y axis and R_z around the z axis.

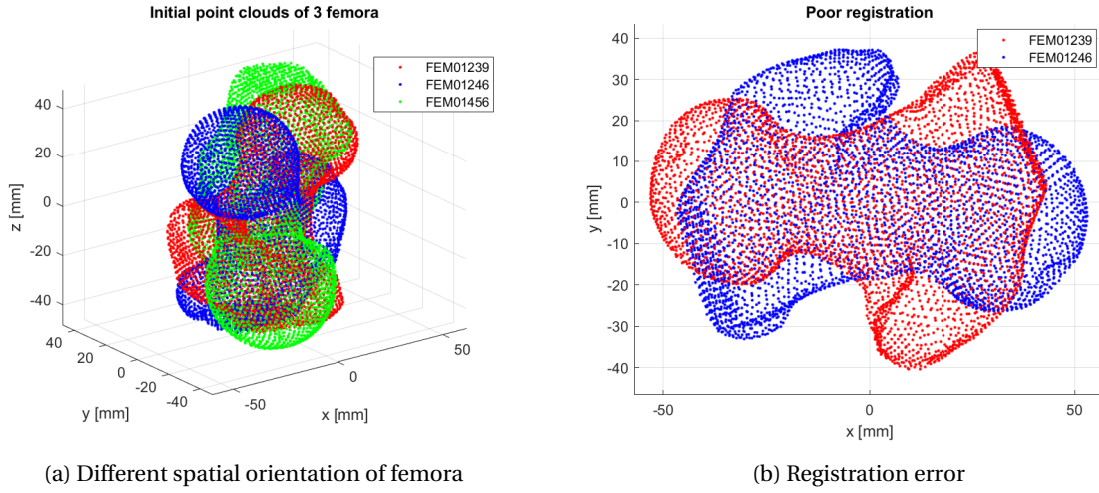


Figure 5.1: Result of registration using PCA

After all this, the final result of these algebraic operations can be seen in figure 5.2 where all femora share the same coordinate system.

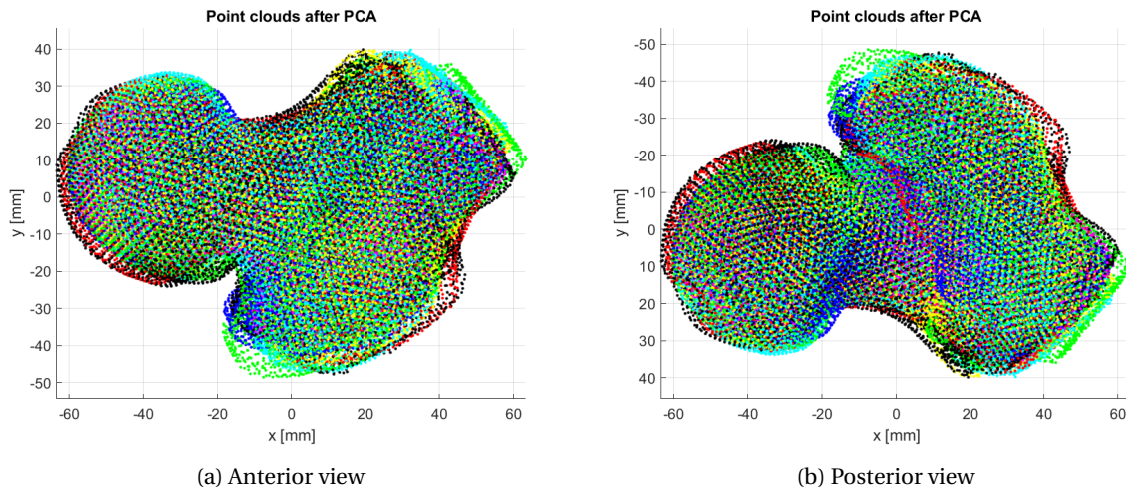


Figure 5.2: Result of registration using PCA

From figure 5.2, one can observe that the PCA successfully aligned most femora in the database but, if singular cases are observed, some changes in geometry lead to poor results in the registration using this method, reason why the data needed refinement. This is the case of the size of the femoral shaft in the training set explored in 4.3.2. In figure 5.3, the case of femora FEM01407 and FEM01462 is observed, representative of two different sized femora with different shaft geometry. Even though PCA is size invariant, the size of the femur will not impact the principal components, different geometry will as is the case here.

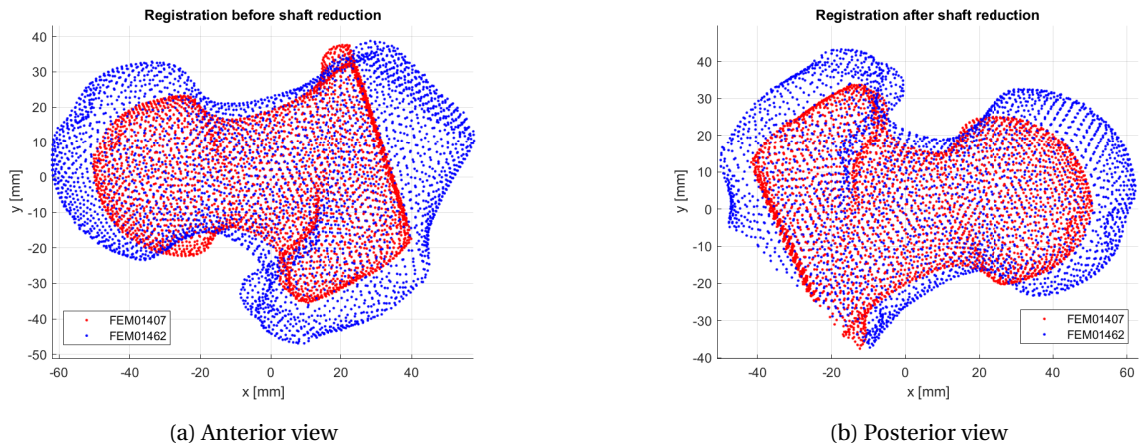


Figure 5.3: PCA registration of femora FEM01407 and FEM01462 before and after shaft removal

After reducing the shaft and reiterating the PCA algorithm, by observing figure 5.3, it becomes clear that the reduced shaft models have a better registration between each other since the principal components of femur FEM01462 better describe the overall shape of the more important regions, the femoral head and neck. This illustrates the overall improvement in the whole set, depicted in figure 5.4.

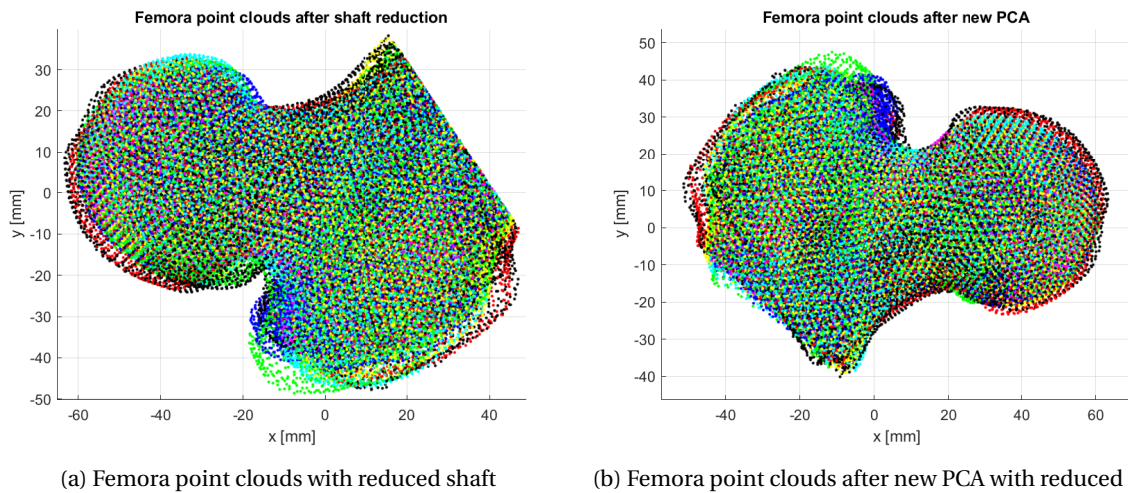


Figure 5.4: Shift of some femora in the point cloud as a result of the shaft reduction

One can observe that the reiteration of PCA lead to the adjustment of some of the femora, thus having a less fuzzy edge all around the model. Furthermore, the region around the top part of the femoral neck and head seems to be better defined as well.

5.1.2 ICP registration

With the coarse registration finished, it is time to discuss the result of the ICP algorithm, responsible for the fine point to point matching of all models in the database. The ICP algorithm demands that one data set needs to be used as a reference, as explained in 4.3.3. To better decide what femur should be used as reference, all femora in the database were used in experiments in order to choose based on the mean value root mean square error (RMSE) between the reference femur and the remaining femora in the database. The result is shown in 5.5.

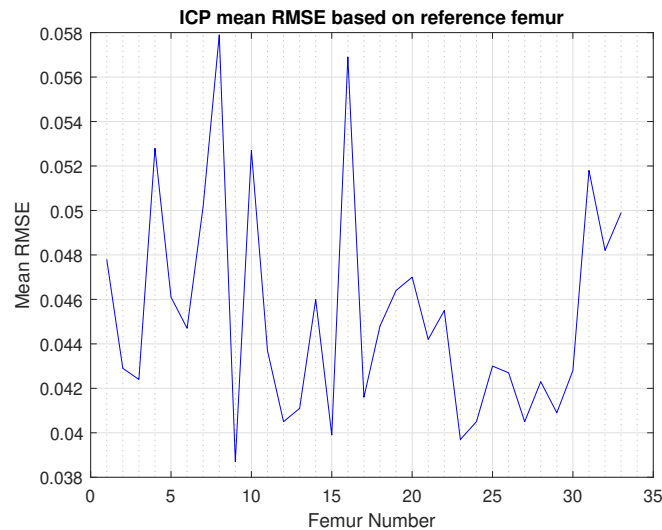


Figure 5.5: Mean RMSE of each femur if used as reference

From figure 5.5, it is safe to say that the femur with the lowest root mean square error is femur 9, FEM01388 from table A.1, with RMSE = 0.0387 mm with a standard deviation of 0.0086 mm, so this will be used as the reference femur for the ICP algorithm. The low root mean square error value may be related to the overall good quality of the database femora as far as point distribution along their surface goes. The final result after applying the iterative closest point algorithm is presented in figure 5.14 from different viewpoints.

Based on the mean RMSE of every femur with the reference one and the geometry presented in figure 5.14, one can say that the ICP algorithm was able to correctly register every femur in the database and successfully generating a good data set to obtain the mean shape of a femur.

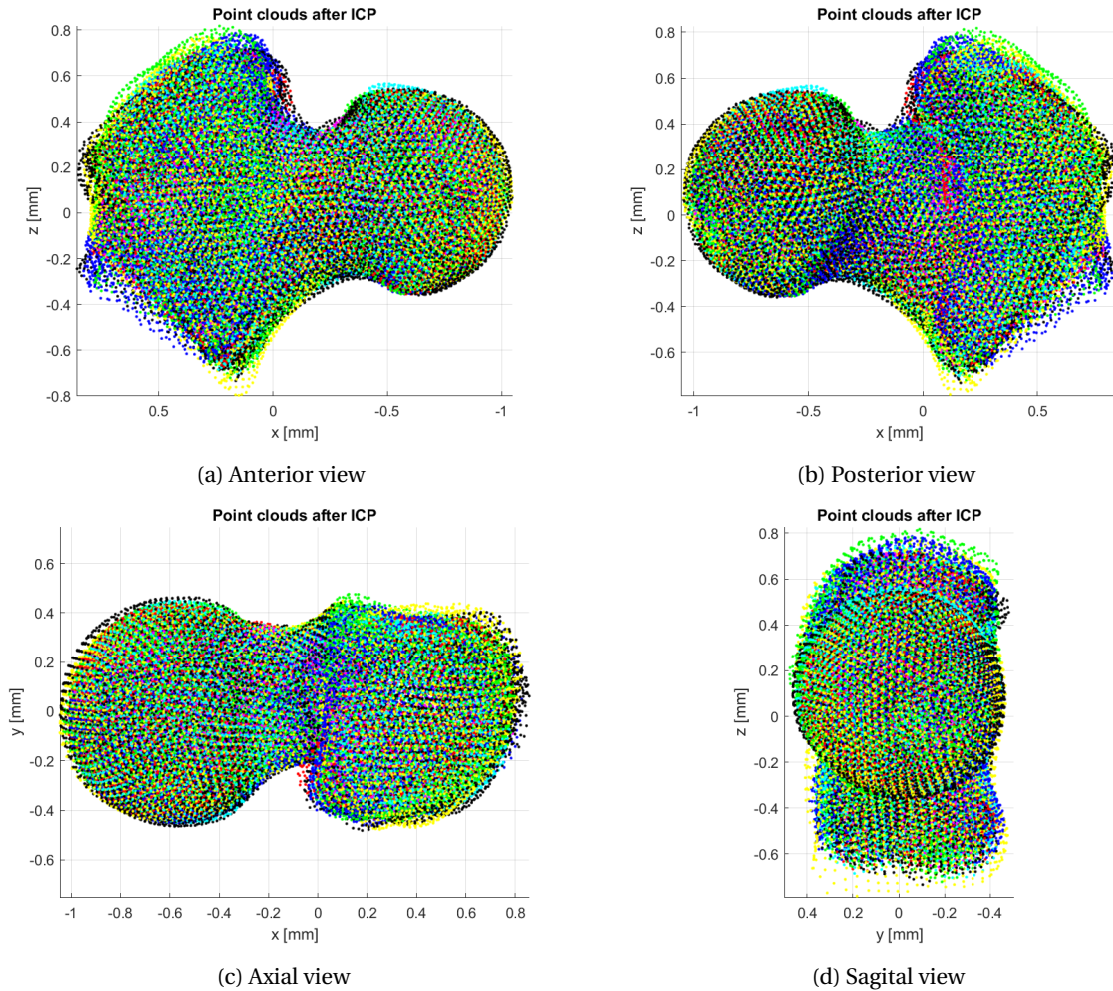


Figure 5.6: Resulting point clouds from ICP registration

5.1.3 Mean Shape model

To obtain the mean shape model of the femur it is necessary to calculate the average value of every point in every femur in the database. This is only possible if the order of the points in the matrices of each femur is the same, this is, two points describing the same location in the femur should be placed. This is achieved by using the remapping algorithm explained in 4.4.

From figure 5.7, it is possible to observe that the maximum number of points from FEM01388 that is lower than the minimum allowed occurs for a grid step of either 1.95 or 2.00. Calculating, results in a downsampling to 3172 and 3148 points, respectively. So, the downsampling of femur FEM01388 was performed using a grid step of 1.95 mm, resulting in a normalized grid step of 0.0344, meaning that in the real size model there is a point in every $1.95 \times 1.95 \times 1.95$ mm box. This resulted in having the number of points in femur FEM01388 reduced from 3354 to 3172 (figure 5.8).

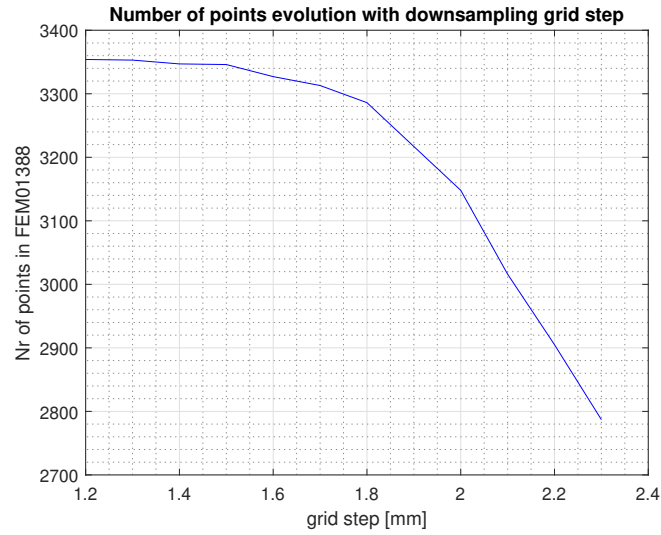


Figure 5.7: Evolution of number of points of femur FEM01388 based on grid step value

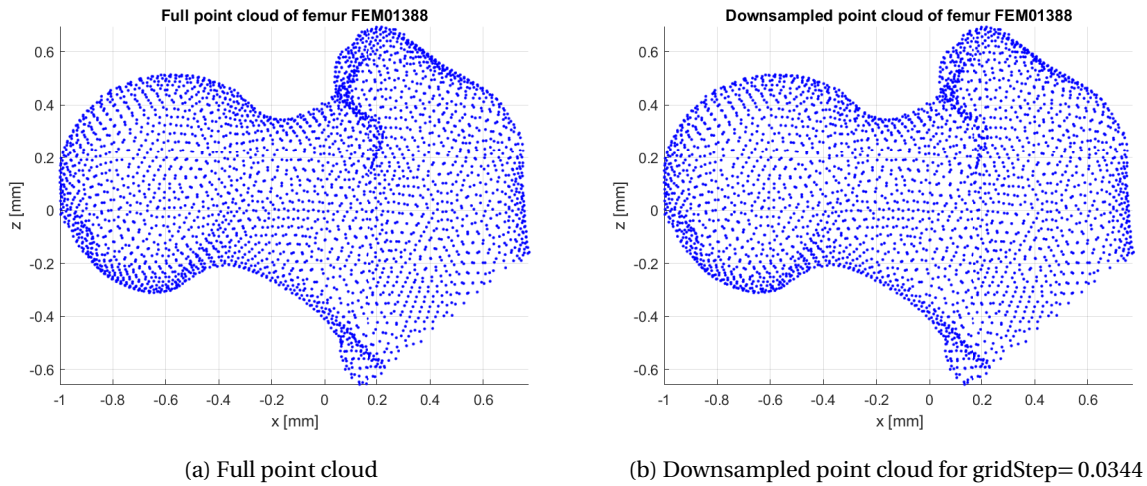


Figure 5.8: Posterior view of femur FEM01543 point cloud before (a) and after downsampling (b)

From figure 5.8, it is not possible to see any difference due the small change in the number of points (5.4%). With the remapping algorithm presented in 4.4, it is now time to remap the points of all femora to the ones from femur FEM01388, denormalize the data and apply equation 4.2 to obtain the mean shape of the femur, presented in figure 5.9.

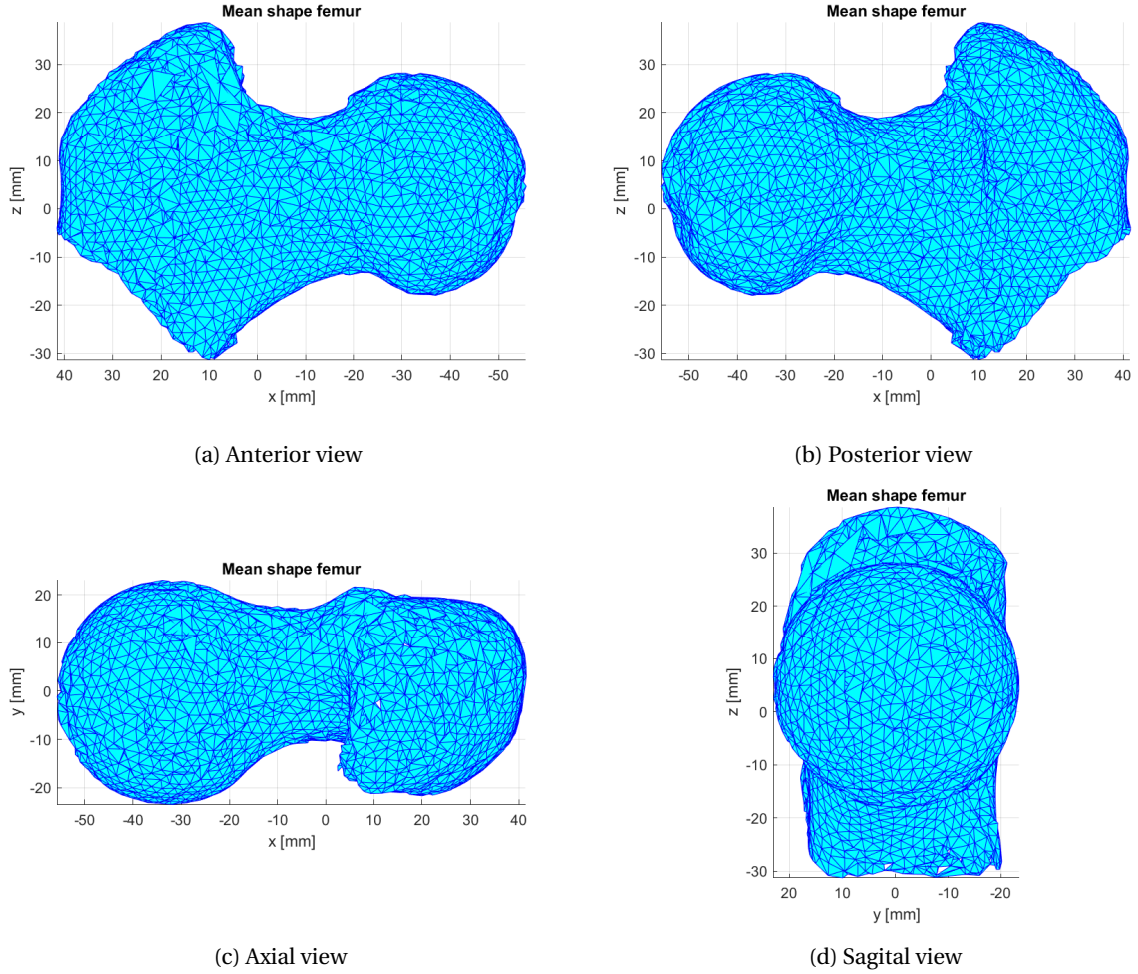


Figure 5.9: Obtained mean shape of the femur

5.2 Statistical Shape model

In order to evaluate the statistical shape model of the human femur it is important to look at the modes of variation obtained through the singular value decomposition on matrix L (shown in equation 4.4) in order to obtain matrices U , with eigenvectors, and D , with the respective eigenvalues λ_i . The SVD algorithm was applied with MATLAB® function `svd` and the aforementioned matrices were obtained.

D is a 9516×33 diagonal matrix with the eigenvalues that represent how much the respective eigenmode influences the model (to note that 4.1 is a single array in the SSM implementation [40], hence $3 \cdot 3172$). To determine the t most relevant eigenmodes, each eigenvalue was compared to the total sum of all eigenvalues, or total variance, given by $\sum_{i=1}^k \lambda_i$. The cumulative variance, $\sum_{i=1}^t \lambda_i$, was also analysed to observe for what value of t this sum would represent 0.9-0.98 of the total variance, represented in the graph in figure 5.10.

From figure 5.10, 90.57% of total variance is achieved with the first 26 eigenvalues and 97.61% is achieved with 30 eigenvalues. This means that the overall variance in the statistical shape model is captured by at least 26 eigenmodes. Due to the large number of eigenmodes, the modes of variation for the three largest eigenmodes are shown in figure 5.11. Each row shows the variation of the eigenmode between $\pm\sqrt{\lambda_i}$ with $i = 1, 2, 3$. The green femora represent the positive value and the magenta ones represent the negative. By

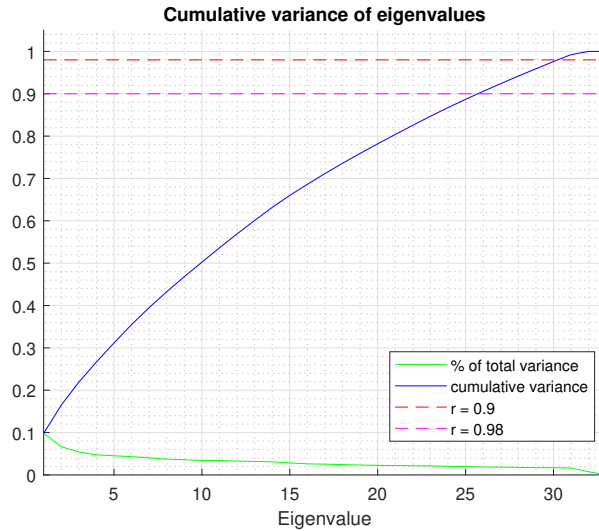


Figure 5.10: Individual and cumulative variance of eigenvalues

observation, it is possible to observe how each eigenmode influences the model's geometry in different ways, such as enlarging or shrinking the entire model (first mode), stretching it (second mode) or enlarging specific areas (third mode). Similar behaviour is found for the remaining 23 modes of variation, although the representation of each mode reduces along the eigenmodes.

The large number of eigenmodes that is needed to represent 90% of the variation in the model may have multiple reasons. Firstly, as a bone, the human femur's geometry is naturally varying. Even though most healthy femora have a similar shape, no two femora are the same so, from the database femora, it is extremely unlikely to observe a pattern of variance that could be represented by a small (less than 10) modes. From [49], for an osteoporotic study, a 2D statistical shape model of the proximal femur was obtained from 506 different people and 95% of the variation in the model was explained by 13 modes. In our case, this is even worsened by the fact that the used models are three dimensional, increasing the complexity of the geometry.

Secondly, from [40], "the power of a statistical model rises and falls with the quantity of available training data", meaning that broadening the database used to obtain the SSM model will inherently produce better results, such as reducing the number of eigenmodes necessary to represent the variation in the model and capture less prominent patterns in the data. The small size of training sets is one of the biggest challenges, mainly due the necessity of optimization when it comes to automatic segmentation algorithms since manual segmentation is specially difficult for three dimensional SSMs.

It is also important to discuss how the point density of each shape in the database affects the overall SSM. One of the biggest drawbacks of Point Distribution models is the necessity of establishing point correspondences between all shapes in the training set. For a 3D statistical shape model this is extremely difficult to rectify if the point density describing the shape is not sufficient. Even the algorithm proposed in this thesis in 4.4 is suboptimal since it will take an already downsampled femur (that has an already altered geometry) as reference and downsample the other femora around it.

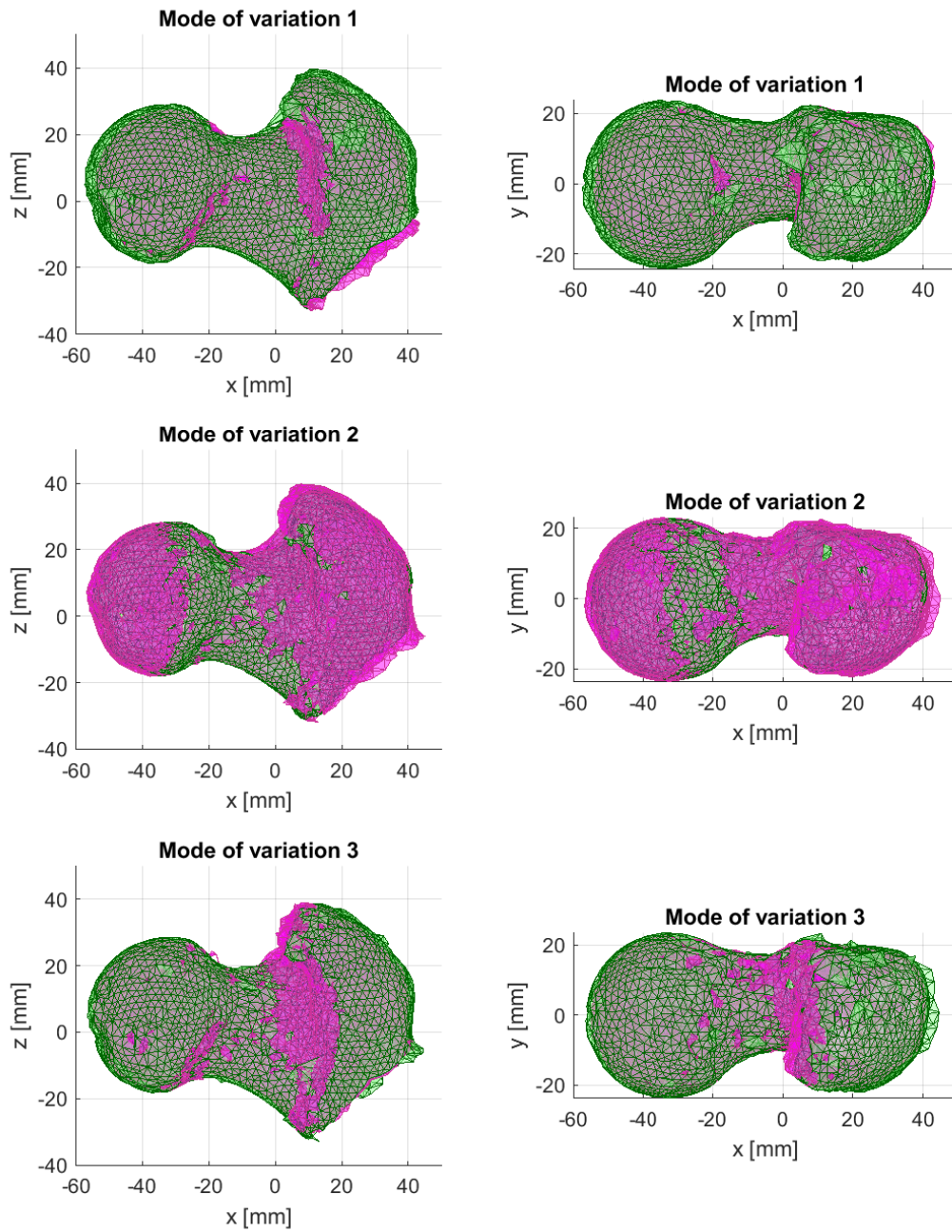


Figure 5.11: First 3 modes of variation of the SSM of the femur in posterior coronal (left column) and axial views (right column)

5.3 Pathological Model and FAI Identification

Identifying a cam type FAI lesion on a patient will start with the acquisition of the pathological femur 3D model. As explained in 4.5, the pathological model was obtained by manual segmentation from 24 radial MRI slices distanced by 7.5° each. This means that one of the challenges will be the fact that the pathological model and the statistical shape model were obtained in different ways. Since the mean shape was acquired based on finite element representations of proximal femora and the pathological femur was obtained by manual segmentation of radial MRI slices, inaccuracies will be introduced into the analysis, since the point distribution between models will differ. Furthermore, the existence of the artifact in every MRI radial slice may have caused minimal flaws in the segmentation of the pathological femur, thus influencing the geometry. As it was explained in 5.2, the way the landmarks are spread across the surface of the femora heavily influences the quality of the results. This becomes clear when observing the point clouds of both models (figures 4.9a and 5.8b). It is then imperative to find the number of points, N , per radial MRI slice that will minimize the point registration error. For this, the ICP algorithm will be used to calculate the root mean square error after the PCA pre-aligns the data. The result of the RMSE depending on the number of points per slice is presented in figure 5.12.

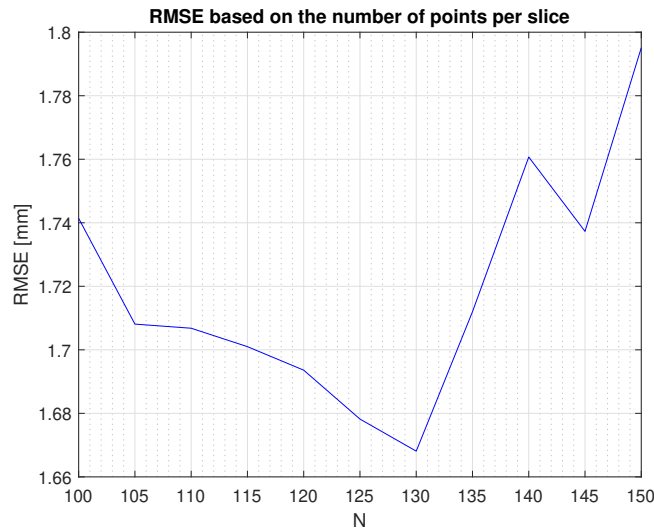


Figure 5.12: Root mean square error variation based on the number of points per radial MRI slice

It is now clear that the optimal number of points per MRI slice is 130 points. The pathological femur also had its shaft reduced in order to improve the PCA pre-alignment, so all points below $z = -65$ mm were removed prior to data centering. The results of the PCA and ICP algorithms are presented in figures 5.13 and 5.14, respectively. The transformation performed by ICP is given by matrix R and vector T , responsible for the rotation and translation.

$$R = \begin{bmatrix} 0.9961 & -0.0332 & 0.0811 \\ 0.0079 & 0.9556 & 0.2945 \\ -0.0873 & -0.2927 & 0.9522 \end{bmatrix}, T = \begin{bmatrix} 10.3736 & 2.5606 & -1.4209 \end{bmatrix} \quad (5.2)$$

RMSE = 1.6681mm

(5.3)

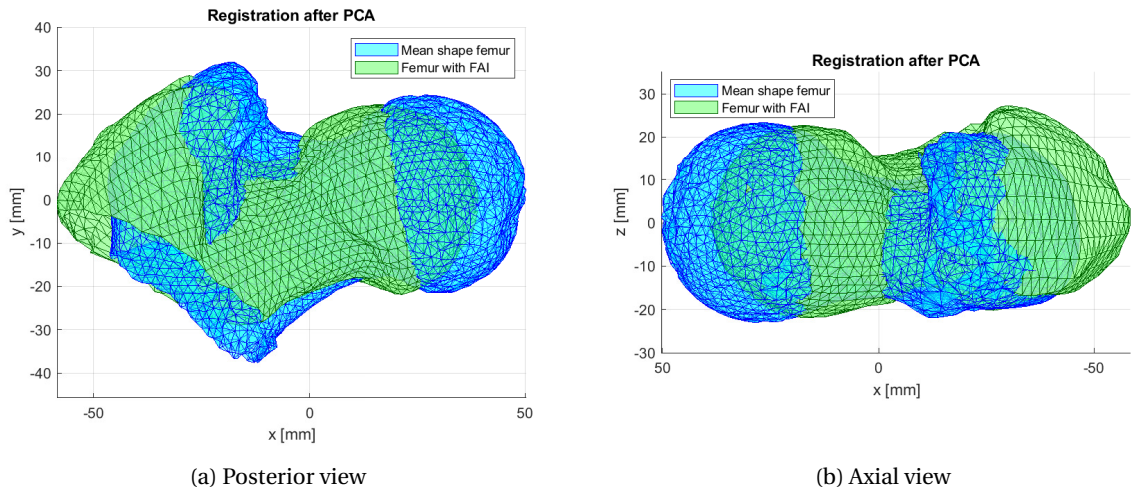


Figure 5.13: Result from PCA registration

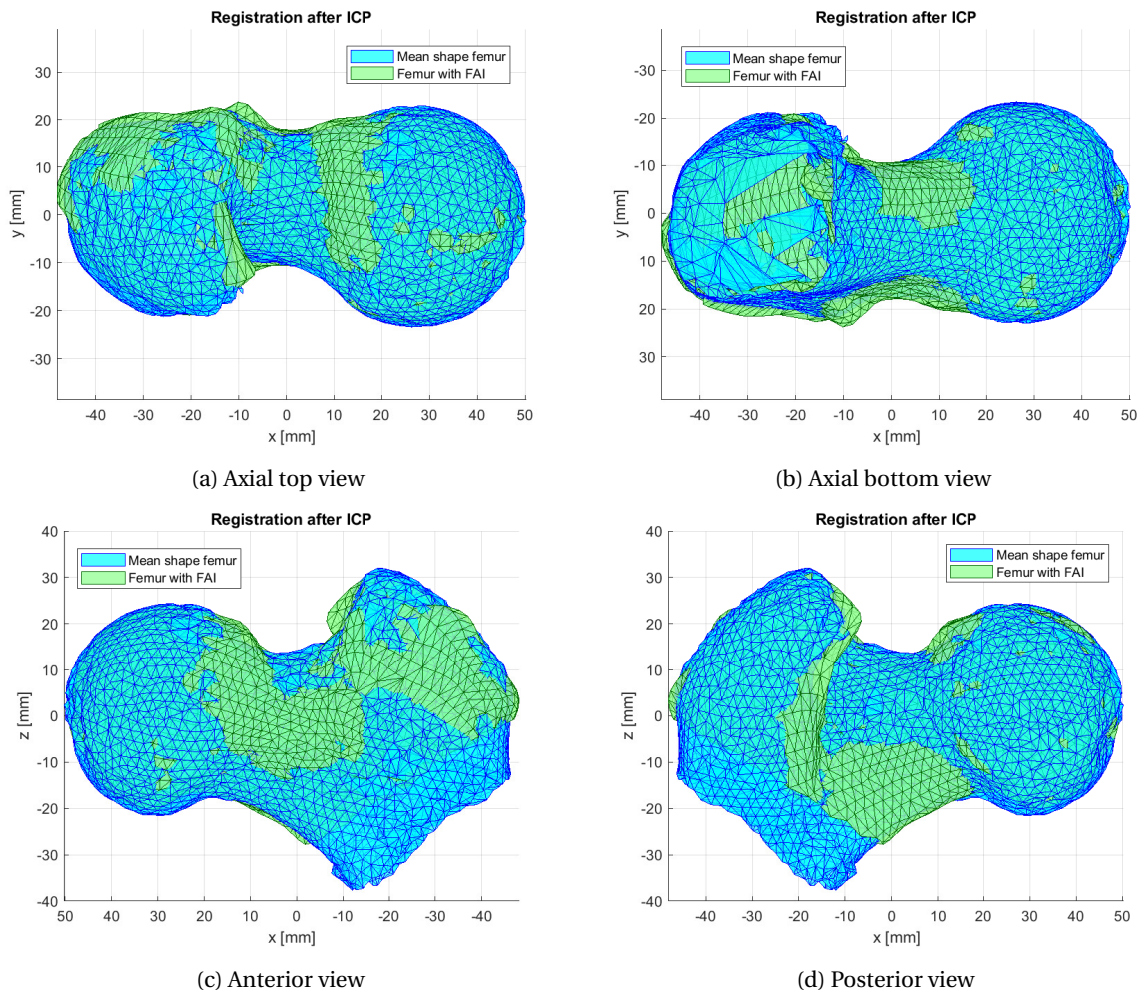


Figure 5.14: Result from ICP Registration

From figure 5.13, it is possible to observe that the PCA algorithm successfully pre-aligned both femora. From the ICP result, in figure 5.14, it is possible to observe regions where both models overlap, and differences in geometry between models can be perceived. From observation, it is possible to clearly distinct the areas where the FAI model (green) has greater volume than the mean shape femur. In figure 5.14a and 5.14c it is possible to distinguish the malformation characteristic of cam type FAI, situated in the anterior/superior region of the femoral head-neck junction.

Regarding the root mean square error obtained for the ICP algorithm, it was considered optimal given the discrepancies in the data related to different point distribution but it is still quite larger than the one obtained during ICP registration for the database femora (0.0387 mm). This is an evidence of how the changes in the point distribution along the model can harm a point to point registration algorithm.

For the mathematical identification of the cam type FAI lesion in the proximal femur it is necessary to compare the pathological femur to the healthy one. Since this comparison would be based on the euclidean distance between points in both models, the points from the mean shape femur were remapped (using 4.4) into the same order as in the pathological model. Remapping the mean shape into the pathological model guarantees that the cam type lesion information is not lost since all points in this femur will have a correspondence to the points in the mean shape. The result of the downsampling is shown in figure 5.15.

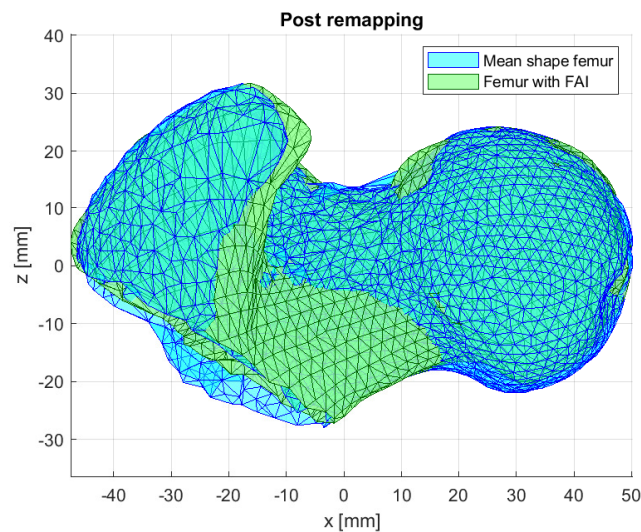


Figure 5.15: Reduction of the mean shape femur to the most relevant areas for the presented FAI femur

It is possible to observe that most of the shaft disappeared from the mean shape femur. This remapping also allows to reduce the number of outliers in order to prepare the data for the identification of the region to be operated.

To mathematically define the area to operate having the statistical model the algorithm purposed in 4.6 was implemented. The mean value of the euclidean distance between the models was 2.1388 mm with a standard deviation of 2.6442 mm. This shows how sparse the data is and how it is not clustered around the mean. This is another evidence of how the point distribution of the pathological femur does not perfectly fit with the mean shape point density. Due this, the remapping algorithm will not correctly create correspondences between points on both models and leave so, some points will be correctly matched while others are

connected to "left over" points, as exemplified in figure 5.16, where points 1 and 2 are correctly matched to points 4 and 5 but point 6 got matched to point 3 that is at a greater distance.

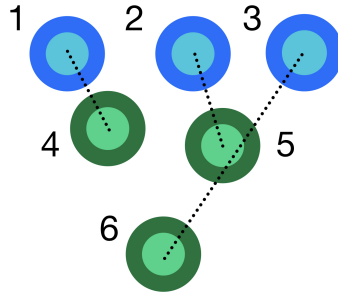


Figure 5.16: Exemplification of how the remapping algorithm is susceptible to changes in point distribution

To improve the results, point 1 of the algorithm presented in 4.6 was repeated and, by eliminating outliers again, the mean decreased to 1.7623 mm and the standard deviation to 1.1676 mm. For the scale of the models, this mean value is not optimal since 1-2 mm in the medical field can be decisive but further elimination of points in the same way would start damaging the geometry of points on the cam type lesion.

In order to define the area of the FAI malformation it was intended to create a surface around the area to be surgically removed. Analytically defining this region proved to be challenging given the impairment caused by the suboptimal point to point matching between the models. The high variation explained by the standard deviation made quite difficult to define an optimal strategy based on the mean value for the distances between models. Even though, visually seems that the models registration was quite successful, the distances do not translate such apparent success. The definition of the region to be removed was then computed by manually selecting clusters of points using MATLAB[®] function `dbscan(X, R, minPts)` that clusters data based on a given minimum of points inside a circle of radius R. As explained in 4.6, it is necessary to find the points of the pathological femur that are at greater distance, x_{up} , from the mean shape as well as the points that are at a distance close to zero, x_{low} , in order to obtain a surface that covers the FAI malformation but has a smooth transition into the normal shaped femur.

Intuitively, it would be expected to find the FAI region by searching the points in the pathological model that are distanced at least the mean value from the statistical femur. In reality, due the large mean value found (1.7623 mm), the threshold was defined at 90% of the mean value to find the largest number of points in the FAI region. To find x_{up} , all points at a distance greater than 90% of the mean were obtained, and from these points, the clusters were generated for $R = 4$ mm and $minPts = 8$. For x_{low} , all points at a distance smaller than 60% of the mean were selected and clusters were generated for $R = 4$ mm and $minPts = 4$. The points and clusters are shown in 5.17.

From the points in these clusters, it was possible to generate a mesh that visually exhibits the region to be surgically removed. Given the manual aspect on the cluster selection it would be needed professional feedback regarding if the area is correctly defined. This region is shown in figure 5.18. Based on the selection of the area with the cam type lesion it was possible to obtain the correspondent points in the statistical femur. These points will define the surface and the shape of the pathological femur after surgery. The final top and

bottom surfaces delimiting the region to remove are shown in figure 5.19.

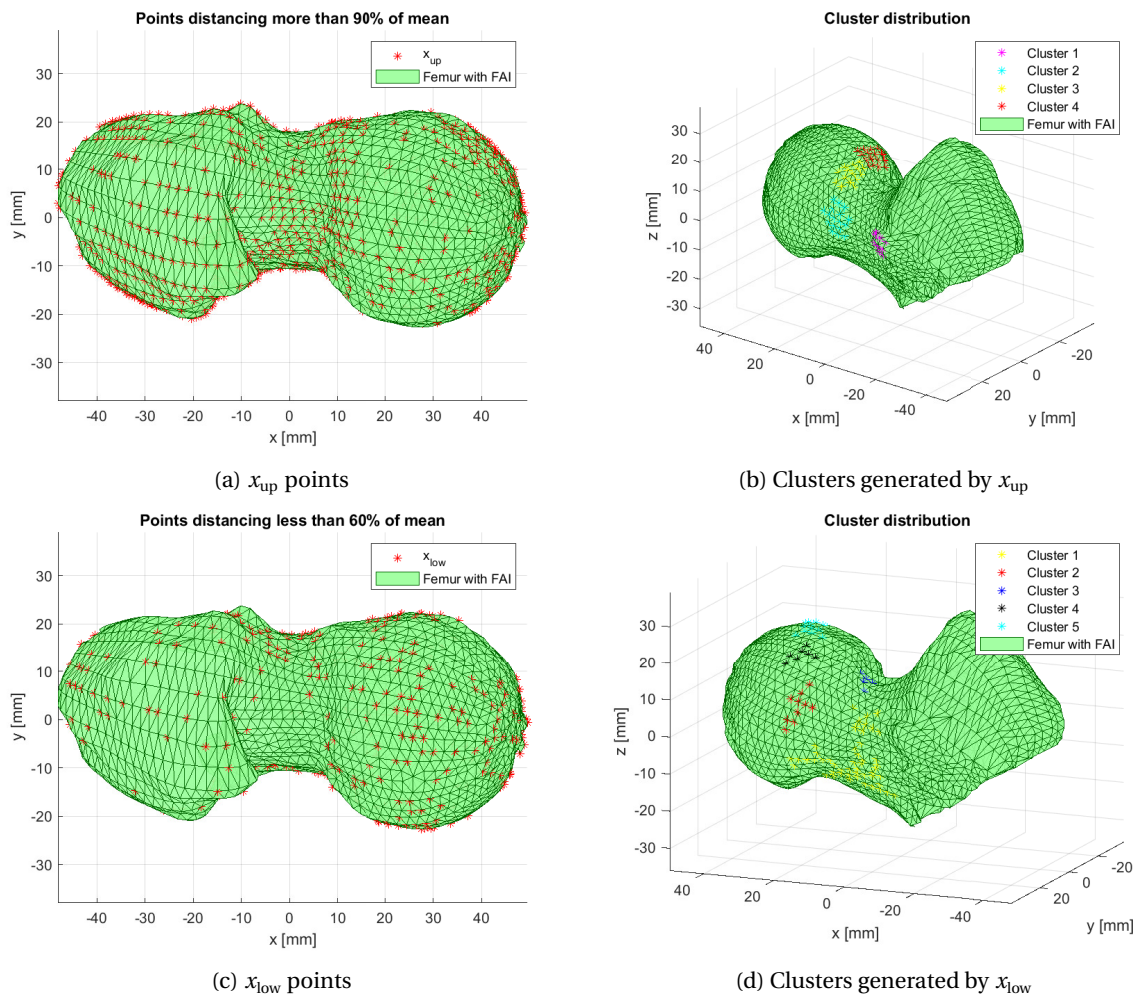


Figure 5.17: Some views of the points at a distance greater than x_{up} ((a) and (b)) and below x_{low} ((c) and (d))

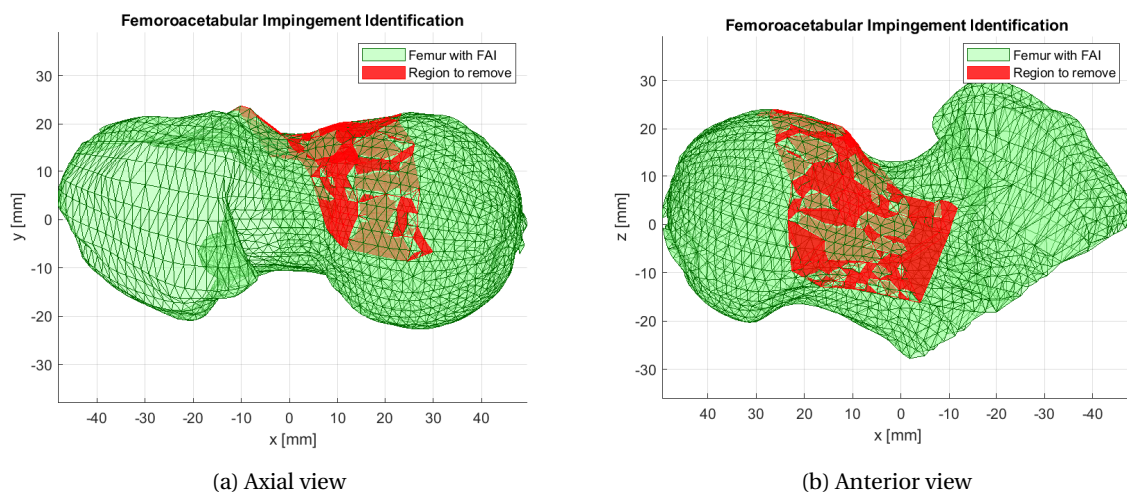
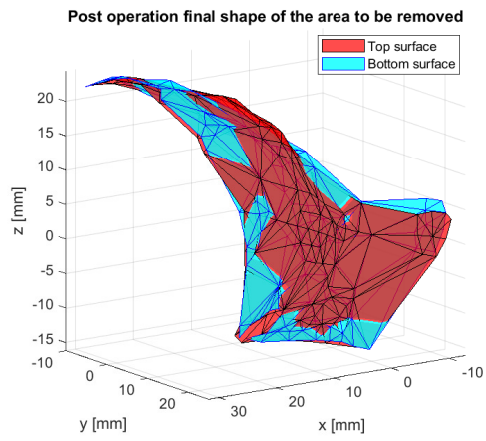
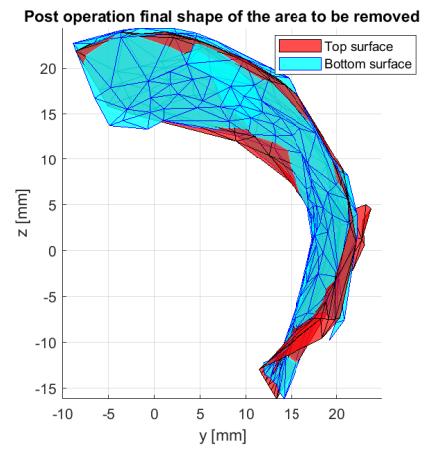


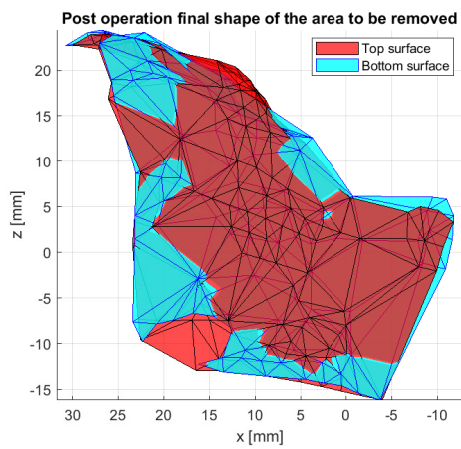
Figure 5.18: Some views of the detected region to be removed from the FAI femur



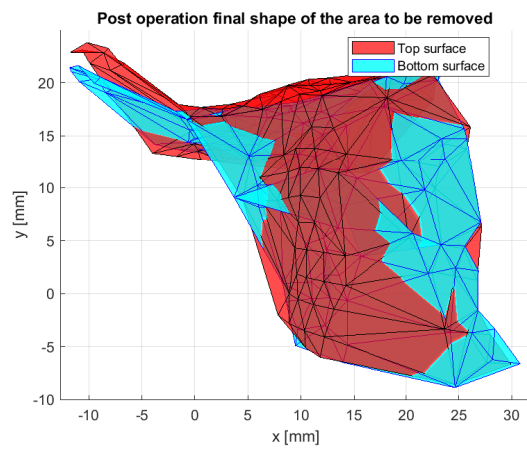
(a) Perspective view



(b) x axis view



(c) y axis view



(d) z axis view

Figure 5.19: Some views of the top and bottom limits of the region to remove from the FAI femur

Chapter 6

Conclusions

The main goal of this thesis was to obtain a 3D SSM of the proximal femur and compare it to a pathological femur in order to identify cam type FAI lesions to be implemented in a developing medical visualization software, Curan.

One of the challenges faced during the development of this thesis was the complexity of the Curan software, for someone with low experience with a full C++ based software, such as myself, specially the underlying architecture. However, given the time put into learning C++ and the cooperation with the project team, it felt natural to implement the FAI Preoperative Planning window and a manual segmentation algorithm in which the user can define the contour in an image coming from a DICOM file series.

Another problem faced was that previous work conducted in the SRL regarding Statistical Shape Models, featured a single 2D SSM from a slice aligned with the coronal plane. This is an oversimplistic approach since the cam type lesions occur over a three dimensional surface along the femoral head and neck, reason why this thesis aimed to obtain a 3D SSM in order to more extensively identify this pathology.

The VPHOP database proved to be a valuable resource since it provided a 3D mesh of 33 different proximal femora. The methods used to obtain the 3D SSM of the proximal femur proved to be a reliable approach for the intended goal. Unfortunately, one of the drawbacks mentioned before, is the fact that Point Distribution Models, such as the one used in this paper, as specially hard to work with, specially on a 3D approach, given the necessity of a point to point mapping between the points of the multiple femora. Regardless, it was possible to obtain a 3D SSM of the proximal femur and study its modes of variation.

The model of the pathological femur proved to be one of the biggest limitations in present work due to how it was generated. Performing the segmentation of the FAI patient's femur on an radial MRI sequence leaves the points in each slice to be distanced from each other by an angle of 7.5° . This means that in the region where the pathology occurs has an inferior point density when compared to the one near the axis of revolution. This difference caused the comparison to the SSM in a pure analytical way to be difficult since both models could not be reliably mapped so that the point to point correspondence would be satisfactory.

In conclusion, while not having optimal results, the present work serves as a baseline for the implementation of a 3D SSM on a medical visualization software. It also became clear how important the quality of data is when working with medical data, but also how difficult it is to find it. All in all, in order to provide a more

reliable identification of a condition, such as cam type FAI, by using a 3D SSM seems promising but further research is needed.

6.1 Future Work

Regarding the 3D Statistical Shape Model of the proximal femora and its modes of variation, it is possible to expect a decrease in the number of eigenvectors necessary to describe the shape variation in the model if more femora are used in the training set. Although the VPHOP database has good models of the human femur, it is possible that the shape variation of the SSM can achieve better results if more data is available. Also, it would be beneficial to have femur data of younger people, since the likelihood of pathologies, such as FAI, develop in younger patients is considerable.

One of the major drawbacks for the present work, as previously mentioned, was the fact that the pathological femur data was obtained using a different protocol when compared to the the mesh models present in the VPHOP database. In order to correct this it would be important to explore the implementation of 3D point interpolation on the FAI femur data in order to fill in the gaps between each slice. Other possible option is to obtain a training set composed of healthy femora obtained from radial MRI sequences. Even though this could improve the point correspondence with the FAI femur, it would not only be a time consuming task to segment enough femora to build a database, it would also be quite difficult to obtain such medical imaging of healthy femora. Furthermore, it has to be considered exploring other shape correspondence algorithms since it has made clear that point to point correspondence seems to be a suboptimal algorithm for complex shapes such as 3D models.

In conclusion, it is important to improve the data quality of the femur models, specially the point density in order to obtain better results.

Bibliography

- [1] K. Atesok, D. Galos, L. M. Jazrawi, and K. A. Egol. Preoperative planning in orthopaedic surgery: current practice and evolving applications. *Bulletin of the NYU Hospital for Joint Diseases*, 73: 257, Oct 2015. ISSN 19369719. URL <https://link.gale.com/apps/doc/A436543523/AONE?u=anon~69c28de6&sid=googleScholar&xid=48243b15>. 4.
- [2] S. R. Marecos. Planeamento pré-operatório para cirurgia conservadora da anca. Master's thesis, Instituto Superior Técnico, 2016.
- [3] Femur - prohealthsys. <https://prohealthsys.com/central/anatomy/grays-anatomy/index-10/index-10-2/index-10-3/femur/>. (Accessed on 10/05/2021).
- [4] O. Jones. The femur - teach me anatomy. <https://teachmeanatomy.info/lower-limb/bones/femur/>, November 2020. (Accessed on 10/05/2021).
- [5] O. Jones. The hip joint - teach me anatomy. <https://teachmeanatomy.info/lower-limb/joints/hip-joint/>, January 2019. (Accessed on 10/05/2021).
- [6] Complete image list - prohealthsys - 551 hip joint sagittal section. https://prohealthsys.com/central/anatomy/grays-anatomy/complete_image_list/,. (Accessed on 10/07/2021).
- [7] D. Power, E. Politou, M. Slaymaker, S. Harris, and A. Simpson. A relational approach to the capture of dicom files for grid-enabled medical imaging databases. In *Proceedings of the 2004 ACM Symposium on Applied Computing, SAC '04*, page 272–279, New York, NY, USA, 2004. Association for Computing Machinery. ISBN 1581138121. doi: 10.1145/967900.967959. URL <https://doi.org/10.1145/967900.967959>.
- [8] Image plane module – dicom standard browser. <https://dicom.innolitics.com/ciods/rt-dose/image-plane>,. (Accessed on 10/17/2021).
- [9] C.7.6.2 image plane module. http://dicom.nema.org/medical/dicom/2020b/output/chtml/part03/sect_C.7.6.2.html,. (Accessed on 10/17/2021).
- [10] M. Leunig, P. Beaulé, and R. Ganz. The concept of femoroacetabular impingement: Current status and future perspectives. *Clinical orthopaedics and related research*, 467:616–22, 03 2009. doi: 10.1007/s11999-008-0646-0.

- [11] A. Kassarian, M. Brisson, and W. E. Palmer. Femoroacetabular impingement. *European Journal of Radiology*, 63(1):29–35, 2007. ISSN 0720-048X. doi: <https://doi.org/10.1016/j.ejrad.2007.03.020>. URL <https://www.sciencedirect.com/science/article/pii/S0720048X0700160X>. Hip Joint.
- [12] K. Matsumoto, R. Ganz, and V. Khanduja. The history of femoroacetabular impingement. *Bone & Joint Research*, 9(9):572–577, 2020. doi: 10.1302/2046-3758.99.BJR-2020-0003. URL <https://doi.org/10.1302/2046-3758.99.BJR-2020-0003>.
- [13] Single-leg-stance-1.png (188x410). <https://dynamichealth.je/wp-content/uploads/2016/01/Single-Leg-Stance-1.png>, . (Accessed on 09/28/2021).
- [14] How to master the single-leg squat | coach. <https://www.coachmag.co.uk/bodyweight-exercises/6727/how-to-master-the-single-leg-squat>, . (Accessed on 09/28/2021).
- [15] G. E. Thomas, A. J. Palmer, A. J. Andrade, T. C. Pollard, C. Fary, P. J. Singh, J. O'Donnell, and S. Glyn-Jones. Diagnosis and management of femoroacetabular impingement. *British Journal of General Practice*, 63(612):e513–e515, 2013. ISSN 0960-1643. doi: 10.3399/bjgp13X669392. URL <https://bjgp.org/content/63/612/e513>.
- [16] Femoro-acetabular impingement cam-type - with alpha angle measurements | radiology case | radiopaedia.org. <https://radiopaedia.org/cases/femoro-acetabular-impingement-cam-type-with-alpha-angle-measurements>. (Accessed on 09/28/2021).
- [17] Femoroacetabular impingement - cam lesion | radiology case | radiopaedia.org. <https://radiopaedia.org/cases/femoroacetabular-impingement-cam-lesion?lang=us>. (Accessed on 10/13/2021).
- [18] C. N. Petchprapa, K. S. Dunham, R. Lattanzi, and M. P. Recht. Demystifying radial imaging of the hip. *RadioGraphics*, 33(3):E97–E112, 2013. doi: 10.1148/rg.333125030. URL <https://doi.org/10.1148/rg.333125030>. PMID: 23674783.
- [19] C. Albers, N. Wambeek, M. Hanke, F. Schmaranzer, G. Prosser, and P. Yates. Imaging of femoroacetabular impingement-current concepts. *Journal of Hip Preservation Surgery*, 4:245–261, 10 2016. doi: 10.1093/jhps/hnw035.
- [20] Femoroacetabular impingement fai sydney, nsw | fao surgery randwick. <https://www.sydneyorthopaedic.com.au/hip-fao-surgery.html>. (Accessed on 08/06/2021).
- [21] Femoroacetabular impingement treatment | fai treatment. <https://nsmoc.com/femoroacetabular-impingement-fai-treatment/>, . (Accessed on 10/13/2021).
- [22] Surgical hip dislocation. <https://surgeryreference.aofoundation.org/orthopedic-trauma/pediatric-trauma/proximal-femur/approach/surgical-hip-dislocation#dislocation>, . (Accessed on 10/13/2021).

- [23] P. Skowronek, M. Synder, M. Polguy, D. Marczak, and M. Sibiński. Treatment of Femoroacetabular Impingement with a Mini-open Direct Anterior Approach. *Indian J Orthop*, 51(6):677–680, 2017.
- [24] S. B. Cohen, R. Huang, M. G. Ciccotti, C. C. Dodson, and J. Parvizi. Treatment of femoroacetabular impingement in athletes using a mini–direct anterior approach. *The American Journal of Sports Medicine*, 40(7):1620–1627, 2012. doi: 10.1177/0363546512445883. URL <https://doi.org/10.1177/0363546512445883>. PMID: 22562788.
- [25] A. Fedorov, R. Beichel, J. Kalpathy-Cramer, J. Finet, J.-C. Fillion-Robin, S. Pujol, C. Bauer, D. Jennings, F. Fennessy, M. Sonka, J. Buatti, S. Aylward, J. V. Miller, S. Pieper, and R. Kikinis. 3d slicer as an image computing platform for the quantitative imaging network. *Magnetic resonance imaging*, 30(9):1323–1341, Nov 2012. ISSN 1873-5894. doi: 10.1016/j.mri.2012.05.001. URL <https://pubmed.ncbi.nlm.nih.gov/22770690>. 22770690[pmid].
- [26] 3d slicer image computing platform | 3d slicer. <https://www.slicer.org/>. (Accessed on 10/13/2021).
- [27] A. Rosset, L. Spadola, and O. Ratib. Osirix: an open-source software for navigating in multidimensional dicom images. *Journal of digital imaging*, 17(3):205–216, 2004.
- [28] Osirix dicom viewer | osirix md. <https://www.osirix-viewer.com/osirix/osirix-md/>. (Accessed on 10/13/2021).
- [29] Dyonics plan user manual and faq.pdf. <https://www.smith-nephew.com/global/assets/pdf/products/surgical/sportsmedicine/dyonics%20plan%20user%20manual%20and%20faqs.pdf>, 2014. (Accessed on 10/13/2021).
- [30] J. Wells, J. J. Nepple, K. Crook, J. R. Ross, A. Bedi, P. Schoenecker, and J. C. Clohisy. Femoral morphology in the dysplastic hip: Three-dimensional characterizations with ct. *Clinical Orthopaedics and Related Research*®, 475(4):1045–1054, Apr 2017. ISSN 1528-1132. doi: 10.1007/s11999-016-5119-2. URL <https://doi.org/10.1007/s11999-016-5119-2>.
- [31] M. C. dos Santos. Preoperative planning for conservative robotic hip surgery. Master’s thesis, Instituto Superior Técnico, 2019.
- [32] Cmake. <https://cmake.org/>. (Accessed on 10/14/2021).
- [33] Asio. <https://think-async.com/Asio/asio-1.19.2/doc/>. (Accessed on 10/14/2021).
- [34] Geometric tools. <https://www.geometrictools.com/>. (Accessed on 10/14/2021).
- [35] Glfw: Main page. <https://www.glfw.org/docs/3.3/index.html>. (Accessed on 10/14/2021).
- [36] M. McCormick, X. Liu, L. Ibanez, J. Jomier, and C. Marion. Itk: enabling reproducible research and open science. *Frontiers in Neuroinformatics*, 8:13, 2014. ISSN 1662-5196. doi: 10.3389/fninf.2014.00013. URL <https://www.frontiersin.org/article/10.3389/fninf.2014.00013>.

- [37] About. <http://openigtlink.org/about>. (Accessed on 10/14/2021).
- [38] Vulkan tutorial. https://vulkan-tutorial.com/#page_About. (Accessed on 10/14/2021).
- [39] About skia | skia. <https://skia.org/about/>. (Accessed on 10/14/2021).
- [40] T. Heimann and H.-P. Meinzer. Statistical shape models for 3d medical image segmentation: A review. *Medical Image Analysis*, 13(4):543–563, 2009. ISSN 1361-8415. doi: <https://doi.org/10.1016/j.media.2009.05.004>. URL <https://www.sciencedirect.com/science/article/pii/S1361841509000425>.
- [41] T. F. Cootes, C. J. Taylor, D. H. Cooper, and J. Graham. Training models of shape from sets of examples. In D. Hogg and R. Boyle, editors, *BMVC92*, pages 9–18, London, 1992. Springer London. ISBN 978-1-4471-3201-1.
- [42] S. L. Brunton and J. N. Kutz. *Data-driven science and engineering: Machine learning, dynamical systems, and control*. Cambridge University Press, 2019.
- [43] J. Hazrati-Marangalou. Database of femur samples. 12 2013. doi: 10.4121/uuid:4ae59365-92f0-480b-a899-ade34bc84a00. URL https://data.4tu.nl/articles/dataset/Database_of_femur_samples/12693476.
- [44] B. Zitova. Mathematical approaches for medical image registration. In R. Narayan, editor, *Encyclopedia of Biomedical Engineering*, pages 21–32. Elsevier, Oxford, 2019. ISBN 978-0-12-805144-3. doi: <https://doi.org/10.1016/B978-0-12-801238-3.99990-2>. URL <https://www.sciencedirect.com/science/article/pii/B9780128012383999902>.
- [45] J. Procházková and D. Martišek. Notes on iterative closest point algorithm. 04 2018.
- [46] Y. He, B. Liang, J. Yang, S. Li, and J. He. An iterative closest points algorithm for registration of 3d laser scanner point clouds with geometric features. *Sensors (Basel, Switzerland)*, 17(8):1862, Aug 2017. ISSN 1424-8220. doi: 10.3390/s17081862. URL <https://doi.org/10.3390/s17081862>. 28800096[pmid].
- [47] Register two point clouds using icp algorithm - matlab pcregistericp. https://www.mathworks.com/help/vision/ref/pcregistericp.html#bupmua2-1_seealso. (Accessed on 10/12/2021).
- [48] T. D. Rachner, S. Khosla, and L. C. Hofbauer. Osteoporosis: now and the future. *The Lancet*, 377(9773):1276–1287, 2011. ISSN 0140-6736. doi: [https://doi.org/10.1016/S0140-6736\(10\)62349-5](https://doi.org/10.1016/S0140-6736(10)62349-5). URL <https://www.sciencedirect.com/science/article/pii/S0140673610623495>.
- [49] B. Wise, L. Kritikos, J. Lynch, F. Liu, N. Parimi, K. Tileston, M. Nevitt, and N. Lane. Proximal femur shape differs between subjects with lateral and medial knee osteoarthritis and controls: the osteoarthritis initiative. *Osteoarthritis and Cartilage*, 22(12):2067–2073, 2014. ISSN 1063-4584. doi: <https://doi.org/10.1016/j.joca.2014.08.013>. URL <https://www.sciencedirect.com/science/article/pii/S1063458414012436>.

Appendix A

FemurID Table

Nr	FemurID	Side	Total number of nodes	Number of nodes after shaft reduction	Percentage of initial number of nodes (%)	ICP rmse if used as reference [mm]
1	FEM01098	Right	3754	3179	84.68	0.0478
2	FEM01239	Left	3748	3252	86.77	0.0429
3	FEM01246	Left	3748	3192	85.17	0.0424
4	FEM01281	Left	4011	3488	86.96	0.0528
5	FEM01286	Right	3943	3534	89.63	0.0461
6	FEM01335	Left	3748	3294	87.89	0.0447
7	FEM01341	Right	3634	3268	89.93	0.0502
8	FEM01386	Right	3751	3411	90.94	0.0579
9	FEM01388	Right	3963	3354	84.63	0.0387
10	FEM01390	Right	3401	3292	96.80	0.0527
11	FEM01407	Right	3553	3329	93.70	0.0437
12	FEM01456	Left	3911	3257	83.28	0.0405
13	FEM01457	Right	3969	3371	84.93	0.0411
14	FEM01458	Left	3962	3382	85.36	0.0460
15	FEM01459	Right	3994	3331	83.40	0.0399
16	FEM01460	Left	3962	3393	85.64	0.0569
17	FEM01461	Right	3938	3319	84.28	0.0416
18	FEM01462	Right	3825	3172	82.93	0.0448
19	FEM01543	Left	3938	3312	84.10	0.0464
20	FEM01544	Left	3861	3190	82.62	0.0470
21	FEM01545	Right	4116	3547	86.18	0.0442
22	FEM01546	Left	4116	3548	86.20	0.0455
23	FEM01547	Right	3855	3278	85.03	0.0397
24	FEM01548	Left	3855	3270	84.82	0.0405
25	FEM01549	Right	3823	3236	84.65	0.0430
26	FEM01550	Left	3823	3243	84.83	0.0427
27	FEM01551	Right	3889	3264	83.93	0.0405
28	FEM01552	Left	3889	3294	84.70	0.0423
29	FEM01553	Right	3952	3278	82.94	0.0409
30	FEM01554	Left	3952	3279	82.97	0.0428
31	FEM01565	Right	4070	3445	84.64	0.0518
32	FEM01566	Right	4070	3454	84.86	0.0482
33	FEM01567	Right	4070	3409	83.76	0.0499

Table A.1: Information from each femur in the database