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

Rafael Fataça
rafaelfataca@tecnico.ulisboa.pt

Instituto Superior Técnico, Lisboa, Portugal

December 2021

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, SSM, FAI, Registration

1. Introduction

Medical imaging software are tools that help a health professional, or researcher, visualizing a set of medical images describing a certain pathology with functionalities that allow a better understanding of the clinical case. Given the necessity of having an open source medical visualization tool that also enables communication with external peripherals, the Curan project was initiated in the Surgical Robotics Lab (SRL), in Instituto Superior Técnico. 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 surgical planning tools had already been developed, more specifically a 2D SSM of the human proximal femur for cam type FAI preoperative planning [1]. 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 indentified when comparing to the SSM. In this work, 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.

2. Background

2.1. 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. [2].

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, Image Position (Patient), S_{xyz}, 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), are the direction cosines of the first row and the first column, respectively X_{xyz} and Y_{xyz}, with respect to the patient and 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),Δ_i, and in the column direction, Δ_j, (second) in mm [3] that allow to calculate

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

where P_{xyz} are the coordinates of voxel (i, j) in the frame's image plane in mm and (i, j) are the column and row indexes, respectively, to the image plane.

2.2. Cam Type FAI and radial MRI sequences

Femoroacetabular impingement (FAI) is a pathomechanical condition that affects the normal functioning of the human hip joint causing abnormal contact stresses in this area [4]. 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.

MRI is widely used for both diagnosis and 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. 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.

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, called a visual artifact. This anomaly darkens the center of all images in a radial series affecting the quality of image in this area.

2.3. State of the Art

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. Some of the State of the Art software tools used for medical visualization are 3D Slicer and OsiriX.

3D Slicer is an open source software platform for analysis and visualization of medical images and for research in image therapy. It 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 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. One of 3D Slicer's drawbacks is the fact that it is not approved for clinical use.

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 [5]. 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.

2.4. 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, [1], examined a way of applying a statistical shape model of a set of healthy femora to compare it with a pathological femur. In this work though, the used data to obtain the statistical femur was a 2D coronal section of each femur. This will be the baseline of the present work but, due to the complexity of human bones, such as the femur in this work, it is expected that a 3D analysis will better depict the characteristics of a real bone.

In [6], the main goal was to develop a preoperative planning tool to be part of a robot-assisted conservative hip surgery. 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 [6] but the segmentation process was implemented independently, so minimal variation on the pathological femur may occur.

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. 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 of the important libraries are: *Asio* used to establish TCP/UDP protocol communication with external applications; *ITK* is a library that provides extensive tools for image analysis; *OpenIGTLink* provides a standardized protocol for the communication between computers and medical devices for image guided therapy. 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 working as a "middleman" between the software and hardware, providing access to the graphics processing units (GPUs). Skia is an open source 2D graphics library that will be used to create and display 2D entities on the screen.

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 FAI pathology. 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.

The SSM 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 here, the segmentation of the pathological femur, used for the results in this work, was obtained in MATLAB® due to trustworthy of results when compared to the emerging Curan software that is still susceptible to changes.

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, some initialization steps are needed:

1. Create a pointer to the thread that is intended to be used;
2. Load the resources necessary for the layout of the Curan app;
3. Load the desired clinical study into memory under the class `curan::display::StudyManager`;
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, `curan::display::DisplayParams` 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 and then `launch_faiviewer_visualizer()` will launch the FAI planning window. At the same time, the `process_os_events` method is called. This method will monitorize all pending events in the created window, processing 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. The FAI window is shown in figure 1. In this figure, it is possible to observe the existence of six buttons in this window. From top to bottom: *Save* stores 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* opens the Segmentation Options Box; *Measure* allows to measure the distance between two points selected in the image.

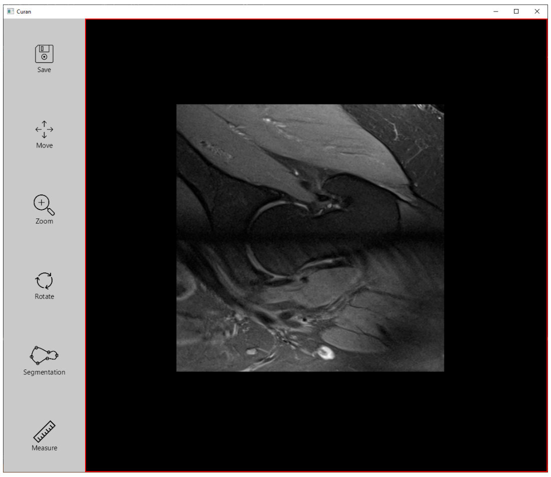

Figure 1: FAI Visualizer in Curan

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. For the time being, it was implemented a way of manually drawing the contour of a region of interest (ROI) in Curan. To start, the user should press the *Segmentation* button in the FAI window. This will open a small menu-like box featuring five different buttons: *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; *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 2). 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.

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 [7]. 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.

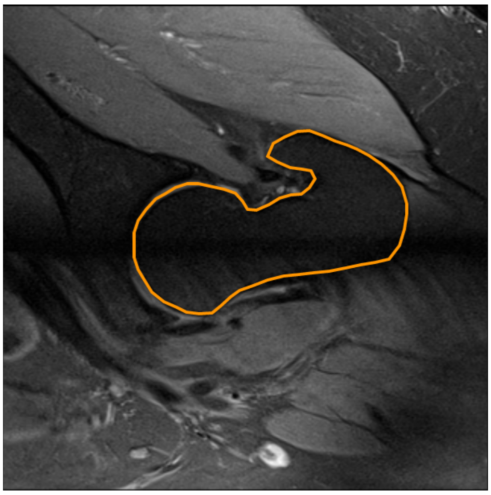

Figure 2: Closed contour drawn in Curan

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 a vector $\mathbf{x}$ [7].

$$\mathbf{x} = (x_1, y_1, z_1, ..., x_k, y_k, z_k)^T \tag{2}$$

A Point Distribuition Model (PDM) of each femur in the training set is then obtained with the landmarks [8]. It is necessary that the landmarks on the different training samples are located at corresponding positions, this is, a landmark point needs to be located in the same relative position in all samples. In figure 3, an example with a resistor model is presented. In order to build a PDM for this resistor it is important to maintain the points in the same position for other resistors 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.

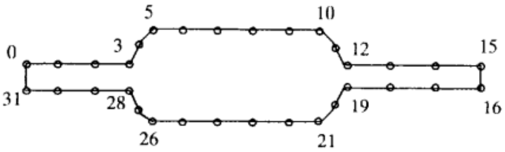

Figure 3: Thirty-two point model of the boundary of a resistor from [8]

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.

$$\bar{\mathbf{x}} = \frac{1}{N}\sum_{i=1}^{N}\mathbf{x}_i \qquad (3)$$

Traditionally, the most significant modes of variation are obtained by eigendecomposition of the covariance matrix of the data. In this work, a Singular Value Decomposition (SVD) will be performed on the aligned landmark matrix L to obtain matrix U, with each column being a mode of variation, and diagonal matrix D with the respective eigenvalue, λ_i [7]. Each column of L is the difference between each femur in the database, and the mean shape, $(\mathbf{x}_i - \bar{\mathbf{x}})$.

$$L = \begin{bmatrix} \vdots & \vdots & \vdots \\ (\mathbf{x}_1 - \bar{\mathbf{x}}) & ... & (\mathbf{x}_i - \bar{\mathbf{x}}) \\ \vdots & \vdots & \vdots \end{bmatrix} \qquad (4)$$

$$\mathbf{x} = \bar{\mathbf{x}} + \mathbf{Pb} \qquad (5)$$

where $\mathbf{P} = (\mathbf{p}_1, \mathbf{p}_2, ..., \mathbf{p}_t)$ is the matrix of the first t relevant eigenvectors and $\mathbf{b} = (b_1, b_2, ..., b_t)$ 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.

$$-3\sqrt{\lambda_i} \leq b_i \leq 3\sqrt{\lambda_i} \qquad (6)$$

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 [7].

4.2. Training Set

The training set used to this work is part of the Osteoporotic Virtual Physiological Human project (VPHOP) and was made public as a research tool by the Eindhoven University of Technology [9]. This database features 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. Each femur information was stored in a matrix with columns having the (x, y, z) [mm] coordinates, necessary to building vector $\mathbf{x}$. Data is then uploaded into MATLAB® R2021a for further analysis.

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 [10]. In this work, the registration will be divided into coarse and fine matching. For the first, Principal Component Analysis (PCA) will be used to perform a pre-alignment by aligning the principal components of each femur to the frame of reference. For the second, the Iterative Closest Point (ICP) algorithm will provide a fine point by point match.

4.4. Principal Component Analysis

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 of the data set. In this work, will be used to perform a coarse matching between training samples by aligning the principal components of each femur to the frame of reference. The first step of PCA is to center the data by subtracting the mean, m_k, to all points X.

$$\mathbf{x} = X - m_k \qquad (7)$$

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

$$C = \mathbf{x}^T\mathbf{x} \qquad (8)$$

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

$$C = VDV^T \qquad (9)$$

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. The femora are then aligned with the frame of reference by multiplying the transpose of V, V^T, by the data.

4.5. Iterative Closest Point

The Iterative Closest Point (ICP) algorithm is one of the most powerful methods to achieve a more precise point cloud matching [11]. 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, 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 [12]. To find matrices R for the rotation and T for the translation, responsible for finding $M^k = \{{M_i}^k | {M_i}^k \in R^3, i = 1, 2, ..., N_M\}$, function `pcregistericp` from the Computer Vision Toolbox in MATLAB® R2021a was used [13].

4.6. Point Remapping and Mean Shape model

Due to different number of nodes characterizing each femur, it is important to downsample the reference 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. Let us take a k ($k = 1, ..., 32$) candidate femur as $\mathbf{X}^k$ an $m \times 3$ matrix and the reference femur as $\mathbf{X}^{\text{ref}}$ 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 $\mathbf{X}^{\text{ref}}$. Finally, the mean shape of the proximal femur is obtained by applying equation 3. The remapping algorithm steps are as follows:

1. For each k femur, compute Δ, an $m \times n$ matrix where,
$$\Delta_{ij} = \|\mathbf{X}_i^k - \mathbf{X}_j^{\text{ref}}\|^2 \qquad (10)$$
with $i = 1, ..., m$ and $j = 1, ..., n$. 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 $\mathbf{X}_j'^k$.
$$\mathbf{X}_j'^k = \mathbf{X}_i^k \qquad (11)$$

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.

4.7. Pathological Femur model

The images of the femur with FAI were obtained at Hospital da Luz using their standard procedure of collecting 24 radial slices. Due to natural variance in color shades in different osseous tissue (cortical and trabecular bone) and the crosstalk artifact, to acquire the pathological femur model, the segmentation will be performed manually, as in [6]. The pathological femur is shown in figure 4.

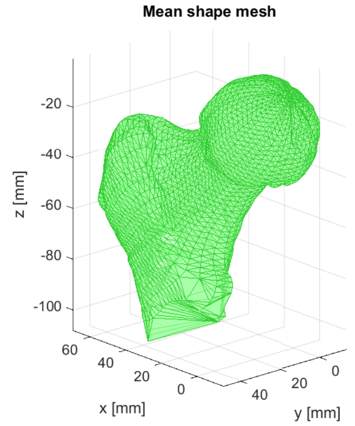

Figure 4: Pathological femur representation

4.8. FAI Identification

It is now necessary to register both femora using the same techniques applied to the femora in the training set, PCA and ICP. 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 will have a higher point density leading to errors in PCA. To correct this, the downsampling function will be used again with the goal of averaging the number of points along the surface of the femur. The remapping algorithm will be once again used to remap points to the same position in both femora matrices and to eliminate points that will not be important for the comparison.

It is known, *a priori*, that cam type FAI deformation occurs on the anterior/superior aspect of the femoral head-neck junction. 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 the mean value and three standard deviations.

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.

5. Results

5.1. Mean Shape Femur

Before diving into the analysis of results it is necessary to discuss the quality of the database. The database features femora models collected from human cadaver proximal femora of older people (77.8±10 years old). 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.

After reducing the shaft and reiterating the PCA algorithm, it turned clear that the reduced shaft models have a better registration between each other since the principal components of each femur better describe the overall shape of the more important regions, the femoral head and neck. These illustrates the overall improvement in the whole set by refining the data and the final PCA result is show in figure 5.

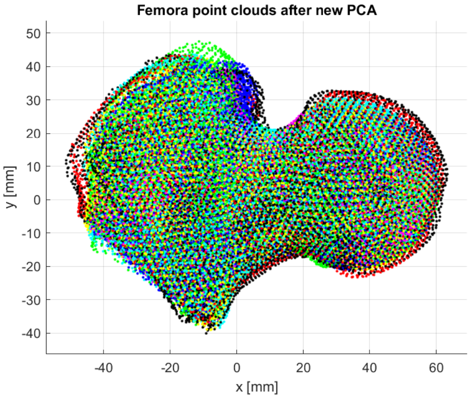

Figure 5: Result of PCA registration with shaft reduced femora

The area around the femoral head and neck seems to be well defined based on the PCA registration. This good results depicts how coherent the data in the database is, meaning that the femora in the training set have a similar point density across their surface.

For the fine matching of the point clouds, the ICP algorithm was used. All femora from the database were taken as the reference femur and femur 9, FEM01388, was chosen as it had the lowest root mean square error (RMSE = 0.0387 mm with a standard deviation of 0.0086 mm) This low RMSE result 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 6.

To obtain the mean shape model of the femur it is necessary to use equation 3 to obtain $\bar{\mathbf{x}}$. The reference femur FEM01388 was downsampled to a grid step of 1.95 mm downsampling it to a number of points lower than all other femora in the database. This will reduce the number of points of this femur from 3354 to 3172 points. After all femora are remapped to the reference femur, the mean shape of the human femur is calculated (figure 7).

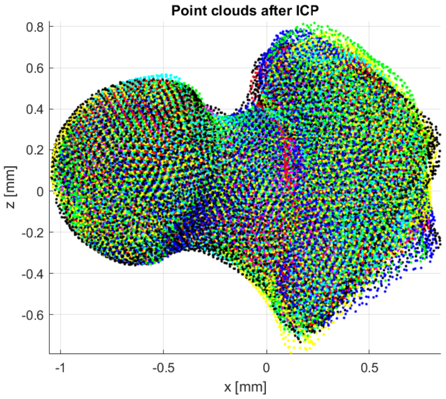

Figure 6: Result from ICP registration

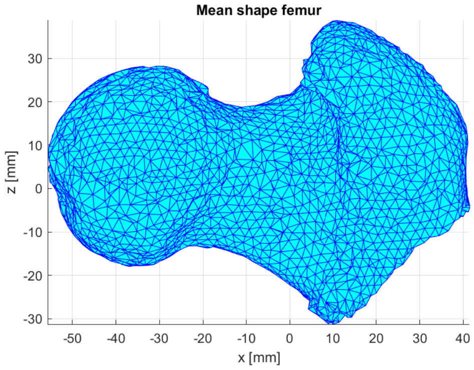

Figure 7: Mean shape of the proximal femur

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. The SVD algorithm was applied with MATLAB® function `svd`. To determine the t most relevant eigenmodes, each eigenvalue, λ_i, 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.

90.57% of total variance is achieved with the first 26 eigenvalues and 97.61% is achieved with 30 eigenvalues. Due to the large number of eigenmodes, the modes of variation for the largest eigenmodes are shown in figure 8. Each plot shows the variation of the eigenmode between $\pm\sqrt{\lambda_i}$. The green femora represent the positive value and the magenta ones represent the negative. By 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) or stretching it (second mode). Similar behaviour is found for the remaining modes of variation, although the variation 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. This is even worsened by the fact that the used models are three dimensional, increasing the complexity of the geometry. Secondly, from [7], "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 also 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 remapping algorithm proposed 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.

5.3. Pathological Model and FAI Identification

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. Due this, 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. To start the identification, both models were registered using the same method as applied for the SSM, by performing a pre-alignment using PCA and a fine matching with ICP. The obtained ICP results are

$$R = \begin{bmatrix} 0.9961 & -0.0332 & 0.0811 \\ 0.0079 & 0.9556 & 0.2945 \\ -0.0873 & -0.2927 & 0.9522 \end{bmatrix} \tag{12}$$

$$T = \begin{bmatrix} 10.3736 & 2.5606 & -1.4209 \end{bmatrix} \tag{13}$$

$$\text{RMSE} = 1.6681\text{mm} \tag{14}$$

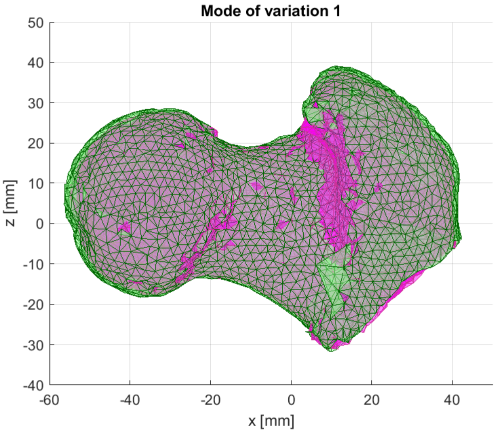

Figure 8: First mode of variation of the SSM of the femur in posterior coronal view

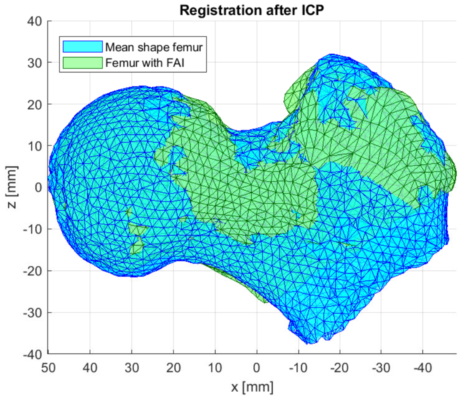

Figure 9: Result from Registration

From figure 9, 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 the anterior/superior areas. Regarding the root mean square error obtained for the ICP algorithm, it is still quite larger that the one obtain 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 into the same order as in the pathological model to remove unnecessary points, such as the ones in the femur shaft area.

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 to some points being well matched while others being connected to a "left over" point. To improve the results, point 1 of the algorithm presented in 4.8 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. 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. 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.

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 10. 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 surfaces is shown in figures 10 and 11.

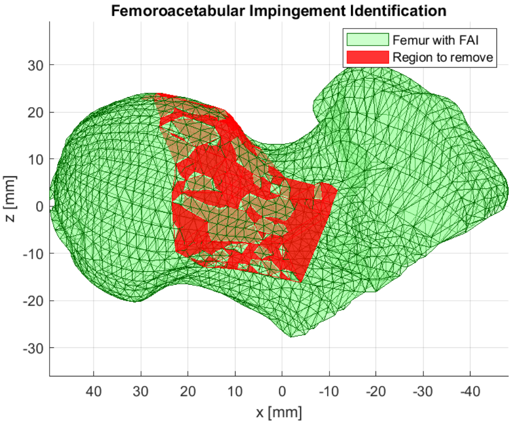

Figure 10: Detected region to be removed from the FAI femur

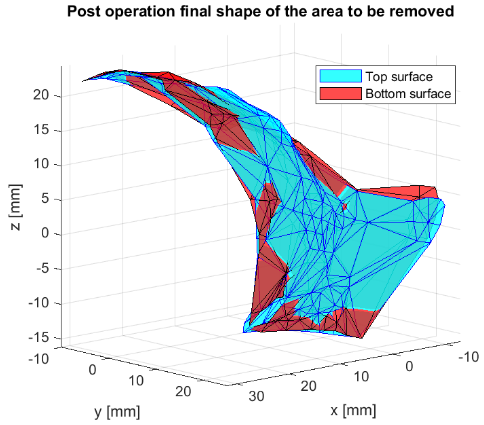

Figure 11: Top and bottom surfaces that limit the region to be removed

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.

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. In future work, it has to be considered exploring other shape correspondence algorithms.

The model of the pathological femur proved to be one of the biggest limitations in present work due to how is 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 will cause an uneven distribution of the points in the model. 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.

In conclusion, while not having optimal results, the present work serves as a baseline for the implementation of how a 3D SSM can benefit pathology identification in 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.

References

[1] Sofia Ruivo Marecos. Planeamento pré-operatório para cirurgia conservadora da anca. Master's thesis, Instituto Superior Técnico, 2016.

[2] David Power, Eugenia Politou, Mark Slaymaker, Steve Harris, and Andrew 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.

[3] Image plane module – dicom standard. https://dicom.innolitics.com/ciods/rt-dose/image-plane. (Accessed on 10/17/2021).

[4] Michael Leunig, Paul Beaulé, and Reinhold Ganz. The concept of femoroacetabular impingement: Current status and future perspectives. *Clinical orthopaedics and related research*, 467:616–22, 03 2009.

[5] Antoine Rosset, Luca Spadola, and Osman Ratib. Osirix: an open-source software for navigating in multidimensional dicom images. *Journal of digital imaging*, 17(3):205–216, 2004.

[6] Mariana Correia dos Santos. Preoperative planning for conservative robotic hip surgery. Master's thesis, Instituto Superior Técnico, 2019.

[7] Tobias Heimann and Hans-Peter Meinzer. Ssm for 3d medical image segmentation: A review. *Medical Image Analysis*, 13(4):543–563, 2009.

[8] T. F. Cootes, C. J. Taylor, D. H. Cooper, and J. Graham. Training models of shape from sets of examples. In David Hogg and Roger Boyle, editors, *BMVC92*, pages 9–18, London, 1992. Springer London.

[9] Javad Hazrati-Marangalou. Database of femur samples. 12 2013.

[10] Barbara Zitova. Mathematical approaches for medical image registration. In Roger Narayan, editor, *Encyclopedia of Biomedical Engineering*, pages 21–32. Elsevier, Oxford, 2019.

[11] Ying He, Bin Liang, Jun Yang, Shunzhi Li, and Jin 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. 28800096[pmid].

[12] Jana Procházková and Dalibor Martišek. Notes on iterative closest point algorithm. 04 2018.

[13] Register two point clouds using icp algorithm. https://www.mathworks.com/help/vision/ref/pcregistericp.html. (Accessed on 10/12/2021).