KTH Royal Institute of Technology

Project Report

CM2006 Medical Image Visualization

Pre-operative planning tool for Total Hip Arthroplasty

Albane Dourdet

Judith Recober

January 6, 2025

Instructors

Rodrigo Moreno

Fabian Sinzinger

Örjan Smedby

Contents

1 Motivation of the project

Total Hip Arthroplasty (THA) is a widely performed surgical procedure aimed at replacing the femoral head with a prosthetic implant. It is typically required when patients experience severe pain and difficulty in daily activities such as walking, sitting, or standing, often caused by arthritis. The most prevalent forms of arthritis leading to THA are osteoarthritis, rheumatoid arthritis, and post-traumatic arthritis. Achieving optimal outcomes from this surgery requires precise preoperative planning. Traditional methods, which rely on 2D imaging and manual estimations by the surgical team, can limit the surgeon's ability to fully understand the patient's unique anatomy or determine the most suitable prosthesis. Advanced tools, such as 3D visualization software, address these limitations by providing detailed segmentations of the patient's femur and hip. This allows surgeons to explore the unique anatomy of the patient interactively, detect anomalies or damages, and plan the procedure with greater accuracy. The software facilitates decisions about the type, size, and position of the implant, ensuring all steps of the procedure are clearly defined in advance. The tool also enhances the surgeon's workflow by transforming preparation steps that were previously manual into efficient, realistic, and digital processes. Features like overlapping and animating implant models with the patient's segmentation provide a realistic preview of implant type, size, and placement. These precise preoperative insights not only optimize surgery time and improve implant positioning accuracy but also reduce the risk of postoperative complications and accelerate recovery times. Furthermore, the software offers significant advantages for surgical teams by fostering collaboration through a shared planning platform. Its intuitive interface makes it easy to adopt without disrupting routine workflows, streamlining the entire planning process. Moreover, the tool does not require specific preprocessing steps, aside from ensuring that the medical images are in .nii format and the prosthesis in .stl format. While primarily intended for orthopedic surgeons specializing in hip procedures, the tool is also a valuable resource for medical students, enabling them to develop and refine their surgical skills through realistic simulations.

Impact analysis : The primary users of this software are surgeons, supported by their surgical teams. Many existing solutions [1] [2] [3] in this domain emphasize the importance of collaborative planning and the ability to simplify measurement processes. Since it was not possible to directly interview surgeons during the design phase of this software, the documentation was based on an analysis of existing solutions and a critical review of their features. This approach took into account the possibility that previous impact analyses of such solutions may not have been comprehensive.

Thus, the analysis of this project is based on a simple scenario and user stories, as "As a surgeon, I want to be able to import the patient's imaging data (CT-scan) directly into the software to visualize the anatomy of the hip and plan the implant." And the software was designed along those user stories, in a simple and intuitive way, in order to fit the surgeons expectations. To meet these expectations, specific design choices were implemented:

- Automatic Region of Interest (ROI) Highlighting: The software automatically highlights the patient's hip, which is the primary region of interest for Total Hip Arthroplasty (THA). Also, it presents all three planes of the patient's 3D images where you can scroll through all the slices, and also use cutting planes from the 3D representation of the hip. This software enables a rapid and comprehensive overview of the patient's body.
- Impact: Surgeons can quickly familiarize themselves with the patient's anatomy, reducing the time spent on identifying key anatomical structures and enhancing efficiency.

These features contribute significantly to improving the preoperative workflow and avoid intra-operative decision-making:

- The app allows the surgeons to accurately plan the surgery, avoiding the need of loosing time during the surgery, making decisions under more stress.
- Enhanced Team Collaboration: The collaborative features of the software (e.g., sharing annotated images with surgical teams) ensure all members of the surgical team are aligned on the surgical plan.

By improving the planning process and reducing intra-operative complications, the software ultimately enhances the patient experience. Specifically:

- Improved Recovery Time: The precise preoperative planning leads to more accurate implant selection and positioning, which contributes to a smoother surgery and faster recovery times.
- Reduced Postoperative Complications: Accurate planning also helps minimize the risk of complications after surgery, ensuring better outcomes for the patient.

Finally, this software can also be used by students to practice choosing and positioning correctly the prothesis. Thus, the software can also significantly contribute to train the surgeon and the future surgeon to enhance the probability of success of the surgery.

2 Description of the project

The developed software provides the surgeon with different tools to get as much information as necessary before starting a surgery.

The app is organized in 4 main windows: three windows where the user can see the slices of the CT images from different planes and be able to scroll, and the third window dedicated to the 3D renderings.

2.1 Dataset

3D CT scans (NIFTI files) of the hip and femur bones that were previously provided in another course for another project (HT2024 3D Image Reconstruction and Analysis in Medicine) are used. Manual ground truth segmentations were also provided for each CT scan. Both images and segmentation masks are used. Moreover, 2 THA prosthesis STL models were acquired from STL finder website [4]. These models were also used in this tool to mimic pre-operative planning steps for a THA surgery.

2.2 Main functionalities and methods

2.2.1 Uploading

Firstly, the software allows the surgeon to interact with the patient's anatomy by loading the 3D segmented data from both the hip and femur. This allows the user to choose the patient data and the respective segmentation mask. In addition, the user can also choose which prosthesis model to upload and on which side it will be placed (either left or right).

2.2.2 Multi Planar Reconstruction of CT scans

There is a specific class that generates a Multi-Planar Reconstruction (MPR) of the CT scans. In one corner the user can see the sagittal slices and scroll over them, in another one the coronal slices and in the other one the axial slices.

vtkImageReslice is used to extract 2D slices from the 3D image data based on the orientation parameter. The set-reslice-orietation method sets the axes based on the orientation parameter. The slices are processed through a vtkImageMapToWindowLevelColors object, to increase contrast and the center of the intensity range. The slices are rendered on a vtkImageActor, and this is added to a vtkRenderer for display.

The class also integrates a QVTKRenderWindowInteractor to enable user interaction with the MPR view within a PyQt user interface. The user can scroll through slices using a slider created by the create-slider method. Then, update-slice updates the displayed slice based on the slider's position. The initial slice display is set to the middle of the volume in the specified orientation using the set-initial-slice method.

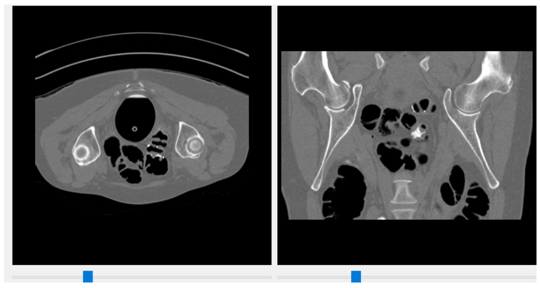


Figure 1: Multi-planar reconstruction views examples

2.2.3 Volume rendering segmentation mask

In the main class, the mask-rendering method is used for the volume rendering of the segmentation mask of the respective image data. vtkGPUVolumeRayCastMapper() is set as the volume mapper. Different properties are defined, such as the volume color with a certain transfer function that sets the background is black and the bone color as almost white. Then, the volume property object is created and the volume color and opacity is connected to this object. Then, volume-property and mapper are connected to the volume actor.

The main interactions related with the volume are zooming and moving around the scene with mouse events.

The opacity-toggle-button is a method that creates a button to be able to change the opacity of the volume. It has both options of seeing the volume fully opaque or semi-transparent, by accessing the volume property.

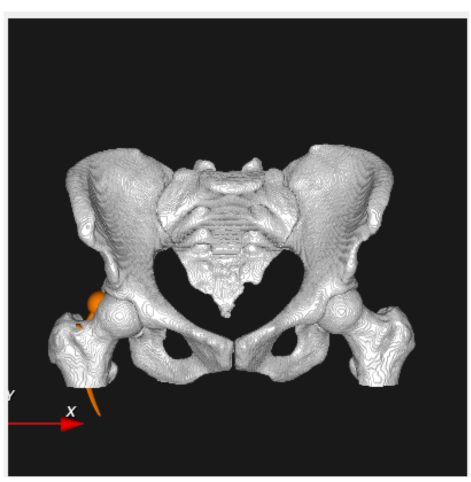


Figure 2: Volume and surface rendering

2.2.4 Surface rendering prosthesis

prosthesis-rendering is the method that renders the surface of the prosthesis STL model. The prosthesis coordinates are normalized to put the model in the same coordinate system as the volume. Since the provided data is an STL file no marching cubes is needed to generate the surface mapper. The actor is defined as well as the properties (orange color and fully opaque). Additionally, a prosthesis transform with different translations and rotations is defined depending on the chosen side by the user.

The main functionalities related with this surface rendering include buttons with which the user can interact to translate, rotate and scale the prosthesis. In the prosthesis-buttons method, different buttons are created and these buttons can change the prosthesis transform by scaling or rotating in the 3 possibles axes by a factor of 5. Every button is connected to the corresponding functionality.

Then, scaling-prosthesis-button method creates 2 buttons, one to scale up and one to scale down the prosthesis, mimicking different prosthesis sizes. The method changes the scale by self.prosthesis-actor.SetScale(new-scale).

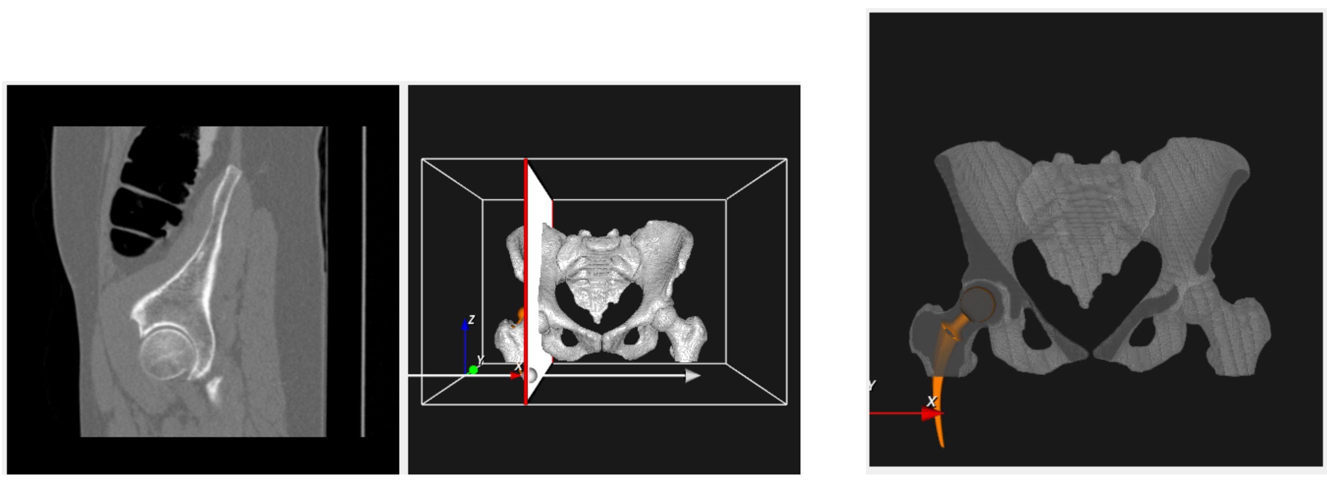
2.2.5 Plane widget

The plane-widget-setup creates a vtkImplicitPlaneWidget instance for interactive plane manipulation and links it to the rendering interactor. The instance is attached to the volume data and sets it to be initially hidden. Then, the GetPlane of the plane-widget is set to a vtkPlane for volume and prosthesis clipping.

To make appear this widget, there is a button that toggles the visibility of the plane widget.

Additionally, there are 2 buttons that enable the user to make a cut in both the volume and surface renderings with the plane of the plane-widget (visually slicing them) with the method RemoveAllClippingPlanes(). Then, the undo-button-setup adds a button to remove the clipping planes and restore the original visualization by resetting any changes made by the cutting plane.

Also, mpr-slice-updates links the plane widget's movements to updates in the MPR views, so when moving the plane widget the slices in the MPR views are adjusted accordingly, ensuring synchronization between the widget and the slices.



(a) Plane widget example

(b) Cutting plane example

Figure 3: Comparison of plane widget and cutting plane examples

2.2.6 Animation mode

The animation functionality of the software enables smooth transitions of the camera's viewpoint, enhancing the visualization experience during pre-operative planning. Indeed, the animate-camera-to-view method animates the camera's position, focal point, and orientation (view-up). This feature allows for dynamic overview of the renderings. The method begins by retrieving the active camera with vtkCamera widget from the renderer. Using VTK's mathematical functionality (vtkMath.Distance2BetweenPoints), the code calculates the distance between the camera and the focal point. This ensures that the camera's movement maintains a smooth zoom effect.

The animation is split into discrete steps, with each step updating the camera parameters through linear interpolation. The position, focal point, and view up vectors are adjusted incrementally, providing a fluid transition. The renderer's clipping range is reset during each update to maintain proper visibility, and the rendering window (vtkRenderWindow) is refreshed to display the updated camera view in real time.

2.2.7 Measurements

The implemented functionality provides tools for measuring distances and angles directly on all 2D slices within the visualization interface. These measurements are essential to assist surgeons in implant selection and positioning.

The distance measurement tool uses the vtkDistanceWidget. The measured distance, initially in pixel units, is converted to millimeters using the GetSpacing() functionality of the associated image data, which provides the dimensions of the reference voxel in millimeters along each axis. Depending on the slice orientation (axial, sagittal, or coronal), the pixel distance is multiplied by the appropriate voxel dimension to yield a meaningful physical distance.

The angle measurement tool uses the vtkAngleWidget, which enables users to measure angles between three points interactively. Like the distance tool, the angle tool dynamically updates and displays the measured angle in degrees in a text actor (vtkTextActor) overlay. Finally, Observers (AddObserver widget combined with InteractionEvent) are used to trigger updates to the displayed values whenever the user interacts with the widgets.

These measurement modes can be toggled on or off using designated buttons in the UI, with visual indicators (e.g., green for the distance tool and blue for the angle tool) to signal the active mode.

Figure 4: Measurement example

3 Results of the project

This software integrates a suite of tools designed to help surgeons in alignment, positioning and prosthesis selection, and collecting essential measurements for pre-operative planning.

The software uses advanced visualization methods, such as multi-planar reconstructions (MPR), volume rendering, and surface rendering, to enable a precise overview of the patient's anatomy together with the chosen implant model. These tools provide surgeons with an intuitive and precise way to interact with the visualized structures. The prosthesis can also be adjusted seamlessly by activating specific controls for each component.

Moreover, cutting planes allow for precise slicing through both the hip and prosthesis renderings. Synchronization between the cutting planes and MPR slices ensures consistency across visualization modes, further enhancing usability. This feature is particularly useful for analyzing how the prosthesis interacts with the underlying bone structure.

Additionally, an animation tool facilitates the transition between different views. This dynamic feature allows surgeons to refine their approach in real-time.

Finally, the measurement tools offer real-time feedback on essential metrics : distances and angles on the 2D views. This integration eliminates the need for external calculation tools, reducing potential errors and streamlining the decision-making process. It also offer a guideline for the OR and a result of the pre-operating process.

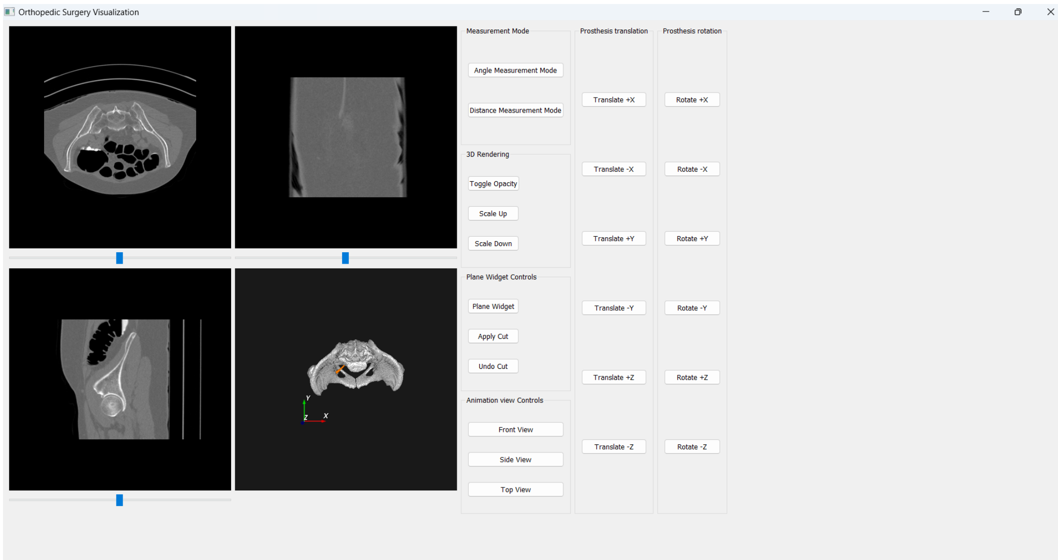


Figure 5: Overview of the Software

4 Conclusions

While the software offers significant advantages, such as enhancing surgical precision, reducing preoperative and intra-operative time, as well as improving team collaboration and the skill development of students and surgeons, there are also limits to consider.

Firstly, since the images and the prosthesis model were acquired from different places, it was hard to put them together in a single space with a proper scale and coordinate system, making it hard to actually mimic real prosthesis sizes.

Moreover, the prosthesis is only visible in the 3D visualization. The implementation of a feature to project the prosthesis position onto 2D slices has not been successfully achieved, limiting the scope of detailed cross-sectional analysis. Capturing the contour of the prosthesis on each slice would have been particularly valuable, as it would allow for a more comprehensive validation of the selected prosthesis's shape. This functionality would provide an additional layer of insight by enabling comparisons of the prosthesis's position not only relative to the bone but also to adjacent tissues such as cartilage, ligaments, and nerves. Such functionality could enhance the surgeon's confidence in the alignment and fit of the prosthesis, ultimately contributing to better surgical outcomes and minimizing potential complications.

Also, the possible movements of the prosthesis are limited and calibrated by the software. To further enhance precision, it could be beneficial to allow users to select the size of the displacement applied to the prosthesis when using the translation and rotation functions.

Finally, a functionality to be able to export data by the user related with the positioning of the implant (information about the measurements and angles) would have been very useful as a guideline during surgery.

Due to time limitations those features could not be implemented. However, the app offers a set of core functionalities, ensuring an effective user experience for analyzing and manipulating 3D medical data for THA implant selection.

References

[1] URL: https://hipinsight.com/.

[2] URL: https://medicad.eu/produkte/3d-2/?lang=en.

[3] URL: https://www.brainlab.com/surgery-products/orthopedic-surgery-products/orthopedic-templating-software/.

[4] URL: https://www.stlfinder.com/3dmodels/hip-replacement/.