Individual Lab Report - 3

Autonomous Reaming for Total Hip Replacement

HIPSTER | ARTHuR

Kaushik Balasundar

Team C:

Kaushik Balasundar | Parker Hill | Anthony Kyu

Sundaram Seivur | Gunjan Sethi

March 3rd 2022

Carnegie Mellon University
The Robotics Institute

Contents

1 Individual Progress

1.1 3D scanning

To convert the model bone to 3D, we approached Prof. Shimada's lab at the department of mechanical engineering. However, a drawback of this was that the laser scanner was static, and could not capture the entire 3D model of the pelvis. However, we realized that this was not necessary to solve the registration problem. The advantage of this method was the high surface resolution and details preserved in the 3D model. We also experimented with the usage of a LiDAR/camera setup using the iPAD pro to get 3D model of the pelvis. However, finer details on the surface of the acetabulum in the model bone was lost in the 3D model. We eventually decided to use the former for this reason since FPFH feature detection is a potential bottle-neck in achieving the required quality of results. Figure 1 shows the various 3D models we acquired for the registration process.

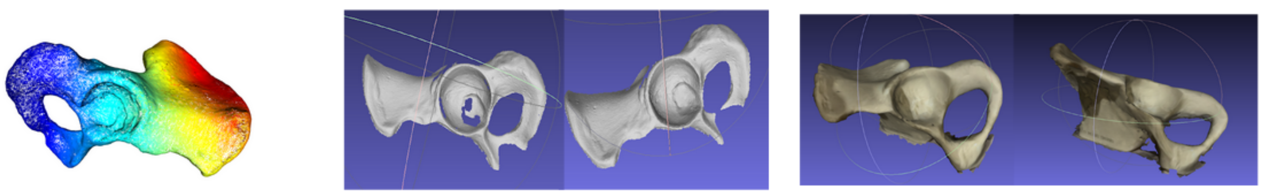

Figure: Off-the-shelf 3D model

Figure: Konica Minolta Vivid 9i

Figure: iPad Pro

Figure 1: Various types of 3D models acquired for registration

1.2 ICP Registration

Leading up to this progress review, I researched various registration techniques that could be implemented for our project. I concluded that the best starting point would be to begin with implementing the ICP registration algorithm since it has remained the industry standard for several years, and also has been proven to work reliably in other medical robotics applications. The requirements of Test 4 involved validating the ability of the ICP algorithm to register the simulated acetabulum surface with the 3D scanned pelvis model. Implementing this required meticulous hyper-parameter tuning and some of the most important ones are listed below:

1. Voxel size: A parameter used to down-sample the target point-cloud before implementing the point-to-point registration method.

2. Distance threshold: A parameter used in the nearest neighbour search when estimating local FPFH features.

3. Initial Transformation Estimate: A rough transformation estimate of how the source point-cloud is related to the target point-cloud.

4. Inlier tolerance: A parameter used to distinguish between points that fit the model and those that are noise and need to be eliminated during the RANSAC-based fine-tuning process.

The result of the hyper-parameter tuning and the registration as a result is as shown in Figure 2 below:

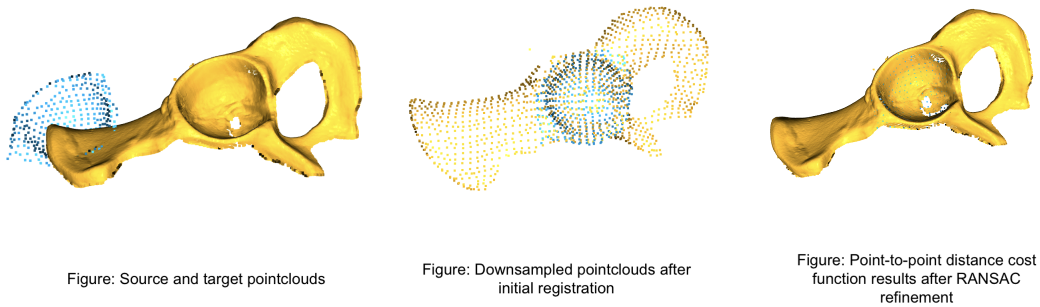

Figure 2: Registration between pointclouds from acetabulum surface and the 3D scanned model of the pelvis

1.3 Simulation

Since the arm we were previously provided did not have the necessary drivers, the simulation environment developed for the arm could not be used. Last week, our sponsors finalized the arm we would be using as the Kinova Gen 3. Following this confirmation, I set up the simulation environment with this arm mounted on a table and imported a pelvis model to be used. We continue to use Gazebo as our simulation environment since existing support was already available for the Kinova Gen 3. An image of the simulation environment developed in Gazebo and the visualization of Moveit! on RViz is shown in Figure 3 below.

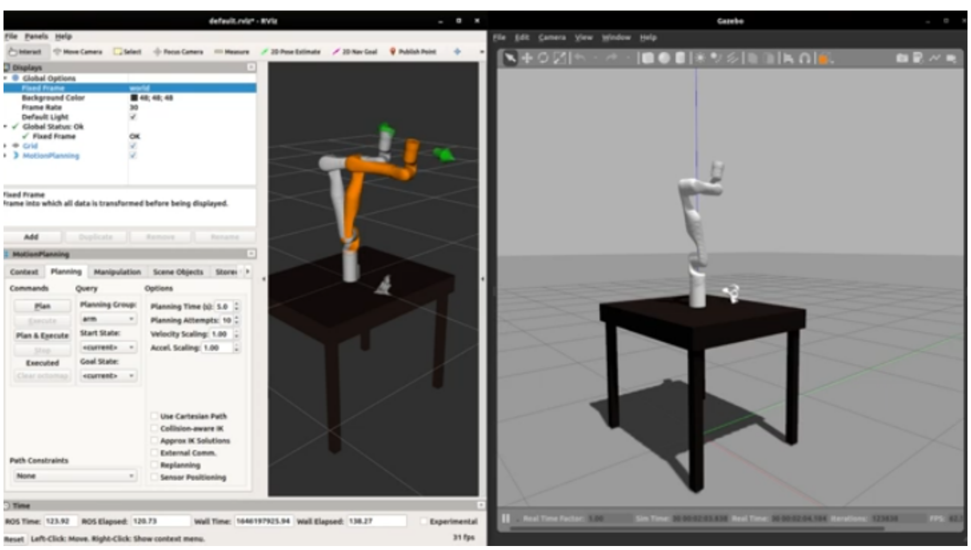

Figure 3: Simulation setup with Gazebo with motion planning in Moveit!

1.4 Marker Pose Transformation

I collaborated with Gunjan in writing the code for acquiring the pose of the marker using the Atycsys camera SDK, converting the rotation matrix to quaternion using the Eigen C++ library and finally broadcasting this transformation as well as publishing the pose through a topic. I helped her conduct the test planned for progress review 2 and document the progress made.

1.5 Presentation

I was also the designated team presenter for this progress review. As a result, I was responsible for ensuring that the tests planned for the review were being planned and executed as per schedule. I also took the lead in structuring the presentation and planning how to communicate it effectively during the progress review.

2 Challenges

2.1 ICP Registration

As mentioned in the earlier section, the most time-consuming aspect of registration involved meticulous hyper-parameter tuning due to the large difference in density between the source and target pointclouds. This disparity is shown in Figure 4. I also undermined the importance of the initial estimate of how the two pointclouds are translated and rotated with respect to each other. I believe that this will be a challenge when working with points acquired from the fiducial markers. As a result, I hope to work alongside Gunjan to acquire this point-cloud as soon as possible so that I have more time to tackle the challenges associated with this process.

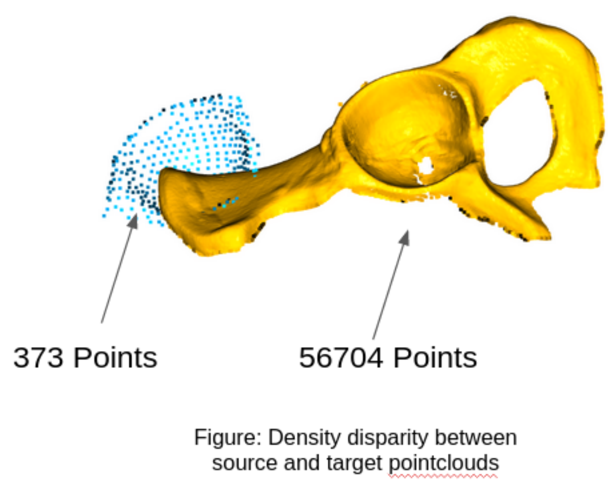

Figure 4: Disparity in density between the source and target pointclouds

2.2 Transform Visualization

Our camera's SDK gives us the rotation matrix and translation vector between the camera frame to the fiducial markers. However, the transform broadcaster using the ROS tf2 package expected a quaternion. There were no native tf2 conversion methods for converting rotation matrices to quaternions and we had to resort to using the Eigen library. However, the Eigen library's output had to be converted back to tf2 compatible message types. In addition, during this implementation, there were several CMake errors that we encountered. Due to our limited experience and exposure with C++ and CMake, figuring out the root cause was time consuming. However, we ensure that each step was documented to save time in the future.

2.3 3D scanning post-processing

As mentioned earlier, the laser scanner we used to obtain the 3D model was static and also had blind-spots causing some areas to have holes. I used the Autodesk MeshMixer software to do post-processing on multiple scanned views and merge the best results. The image below shows the final post-processed mesh geometry.

3 Team Work

The table in Figure 5 summarizes the work performed by various team members:

Sl No	Team Member	Contribution
1	Kaushik Balasundar	I worked on implementing and validating the iterative closest point registration algorithm with the simulated pelvis model. Sundaram and I 3D scanned the pelvis model for the simulation. Once the arm was finalized by our sponsors, I set up the simulation environment with the Kinova Gen-3 arm. I worked with Gunjan in publishing the marker poses to ROS and broadcasting the pose as a TF transform to visualize on RViz. I was the team's presenter for the second progress review.
2	Parker Hill	He helped to set up the physical set-up for the Kinova Gen-3 arm which involved assembling a Vention table, picking up the arm, and setting up the arm on the table. He designed and 3D printed prototypes for attaching the reamer handle to the end-effector. He also worked on creating the conceptual design as well as the schematic of our motor control board for the reamer tool
3	Gunjan Sethi	Gunjan re-calibrated the markers to improve the robustness of the ROS camera node. Further, she added the marker pose detection and visualization features to the node and performed various reliability tests to ensure smooth functioning during the progress review.
4	Sundaram Seivur	Sundaram worked on formulating the optimal control problem and collaborated with Anthony in getting feedback from professors. He studied the functions used to interface the output of the controls loop with ROS. He also worked on setting up the hardware which included getting the arm from our sponsors, assembling the Vention table and mounting the Gen3 arm. Finally, he worked with me to get a 3D of the model bone.
5	Anthony Kyu	Anthony worked on formulating the optimal control problem for the Model-Predictive Controller, and creating several iterations of the optimal control problem. He explored various libraries to use for the MPC controller and for interfacing the controller (in Julia) with ROS. He then started implementing the MPC controller, coding the dynamics function, the constraints and the objective function. He collaborated with Parker on the PCB, discussing requirements and ideas for the board, and researching some components for the board.

Figure 5: Contributions by each team member

4 Plans

In the next few weeks leading up to the progress review 3, I plan to work on the following:

1. **Acquiring 3D point-cloud from the registration marker**: I plan to collaborate with Gunjan in writing the necessary code to capture landmarks from the surface of the acetabulum and converting them to a point-cloud.

2. **Setting up force feedback on the robot arm simulation** : I will collaborate with Anthony in setting up the necessary force control plugins needed to deploy the MPC controller on our robot.

3. **Performing registration on real-world data**: Using the point-cloud that we acquire from the marker, I plan to test out the efficacy of the registration algorithm with real-world data. If this step is done, it will bring us very close to completing the perception pipeline of our project.

4. **Setting up the robot arm with ROS**: I will work alongside Sundaram in setting up the necessary ROS packages and testing out various basic functionalities with the arm. This will serve as the basis of the real-world tests we will deploy with our controller as a part of SVD.