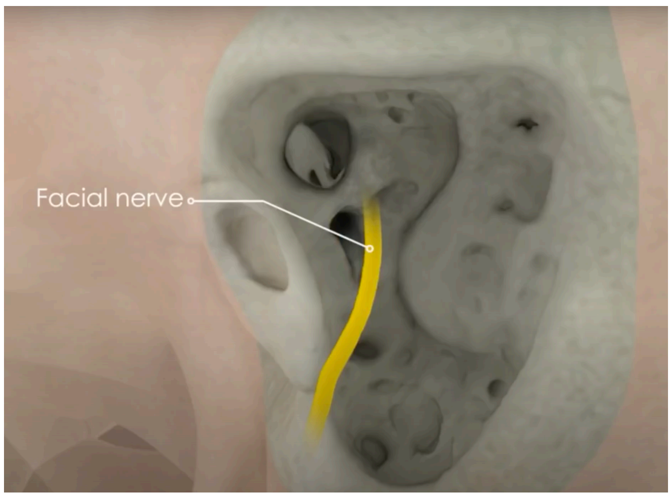
CISii
Final Present 0505:

Anatomical Mesh-Based Virtual Fixtures for Surgical Robots

- Student: Yiping Zheng
- Mentor: Zhaoshuo Li, Russell Taylor
- Group 7, CISii

I. Project Background Overview



Mastoidectomy Surgery (Source: Youtube [5])

$$\begin{aligned} &\arg\min_{\Delta \boldsymbol{q}} \|\Delta \boldsymbol{x} - \Delta \boldsymbol{x_d}\|_2, \\ &\text{subject to } \boldsymbol{A}\Delta \boldsymbol{x} \geq \boldsymbol{b} \\ &\qquad\qquad \Delta \boldsymbol{x} = \boldsymbol{J}\Delta \boldsymbol{q} \end{aligned}$$

Equation(1), Zhaoshuo Li, et al. Anatomical Mesh-Based Virtual Fixtures for Surgical Robots, 2020

II. Previous Work (submitted to IROS 2020 [1])

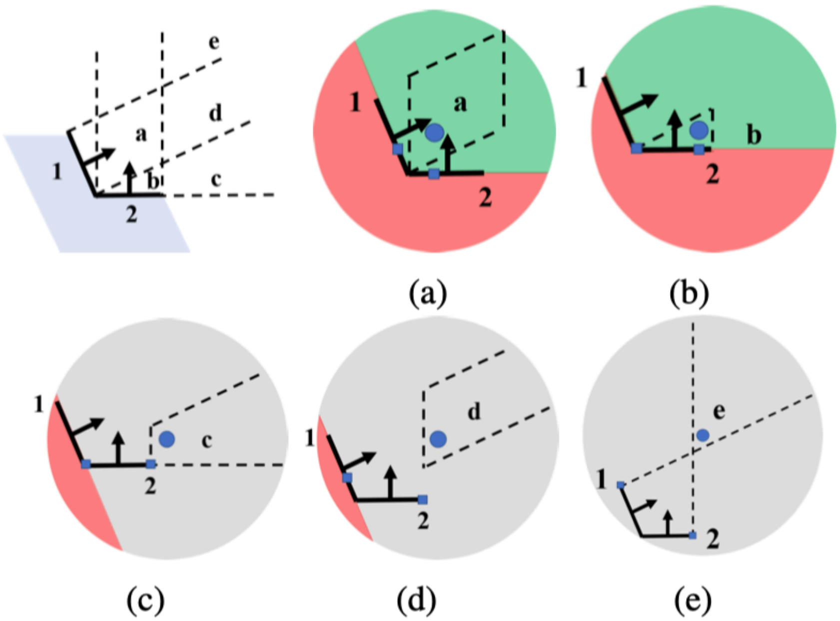


Fig. 5: Enumeration of a **locally concave** shape.

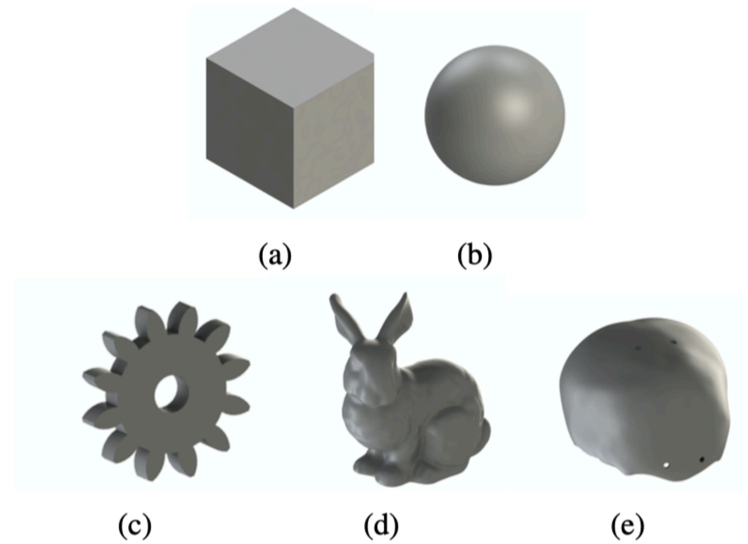
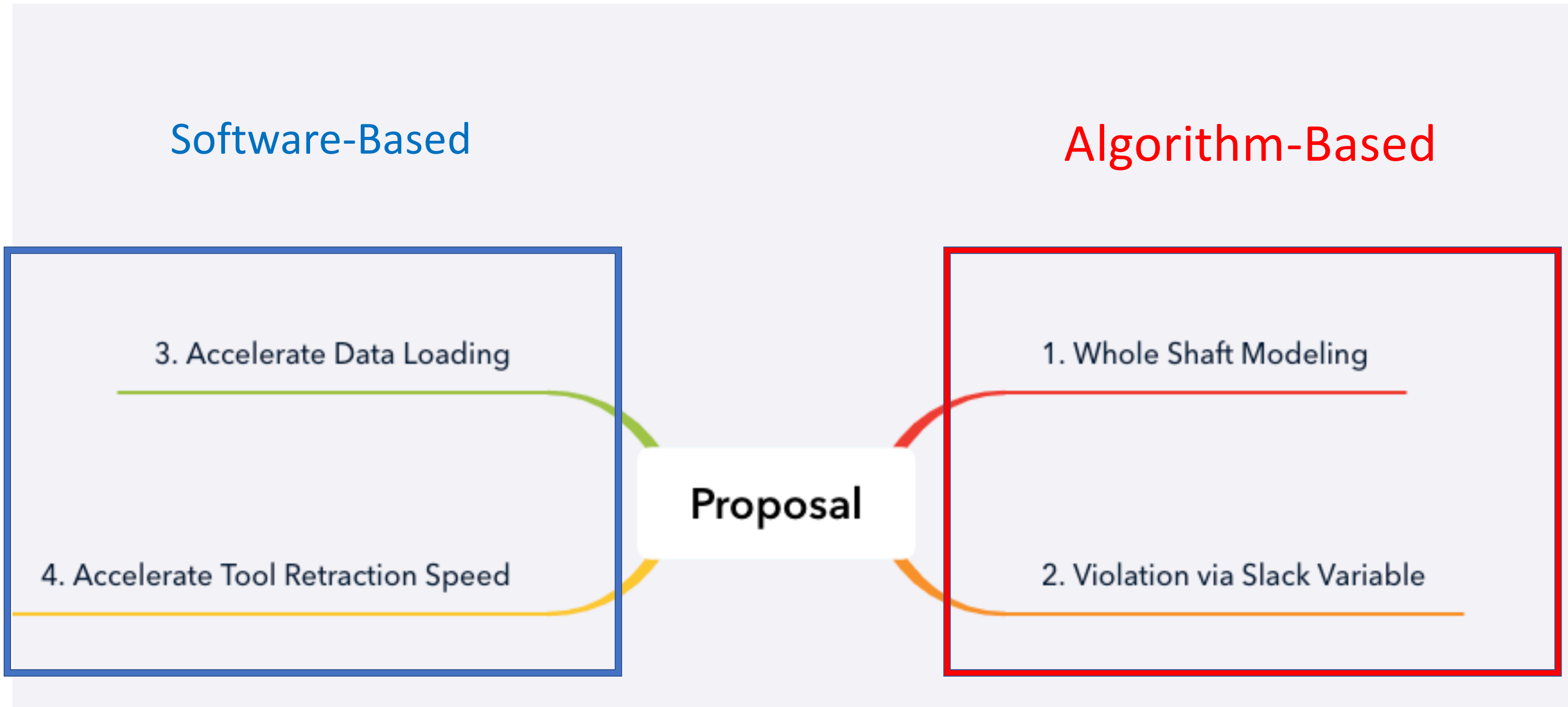


Fig. 6: Examples of the objects tested in simulation. (a) cube, (b) sphere, (c) gear, (d) Stanford bunny, (e) pediatric skull.

Z. Li, R. Taylor, et al. Anatomical Mesh-Based Virtual Fixtures for Surgical Robots [1]

III. The Goal of My project

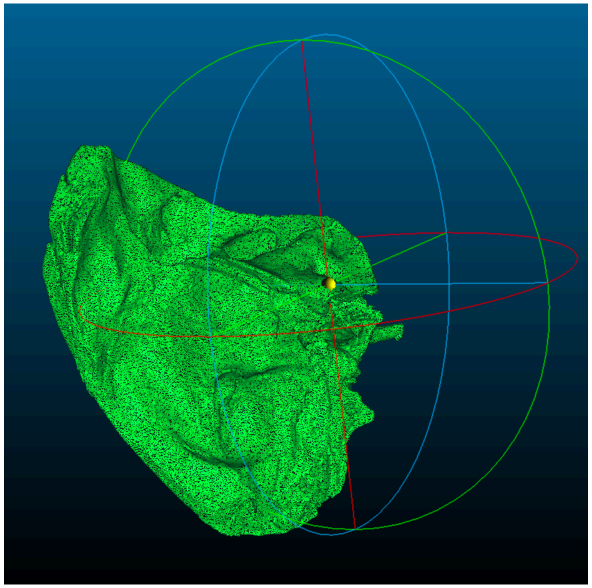


IV. Feature 1: Data Loading Utility Function

- A STL file reader for loading CT scan data to create a cisstMesh data structure.

- It is easier to use
  - Automatic file format detection, either ASCii or Binary.

- It has higher performance:
  - Boosting loading speed by 75%



Eg. Temporal Bone STL, 640,881 points, 213,627 faces. **Binary**: 370MB, 8min10sec to load. **ASCII**: 1.56GB, 31min42sec to load @ CloudCompare

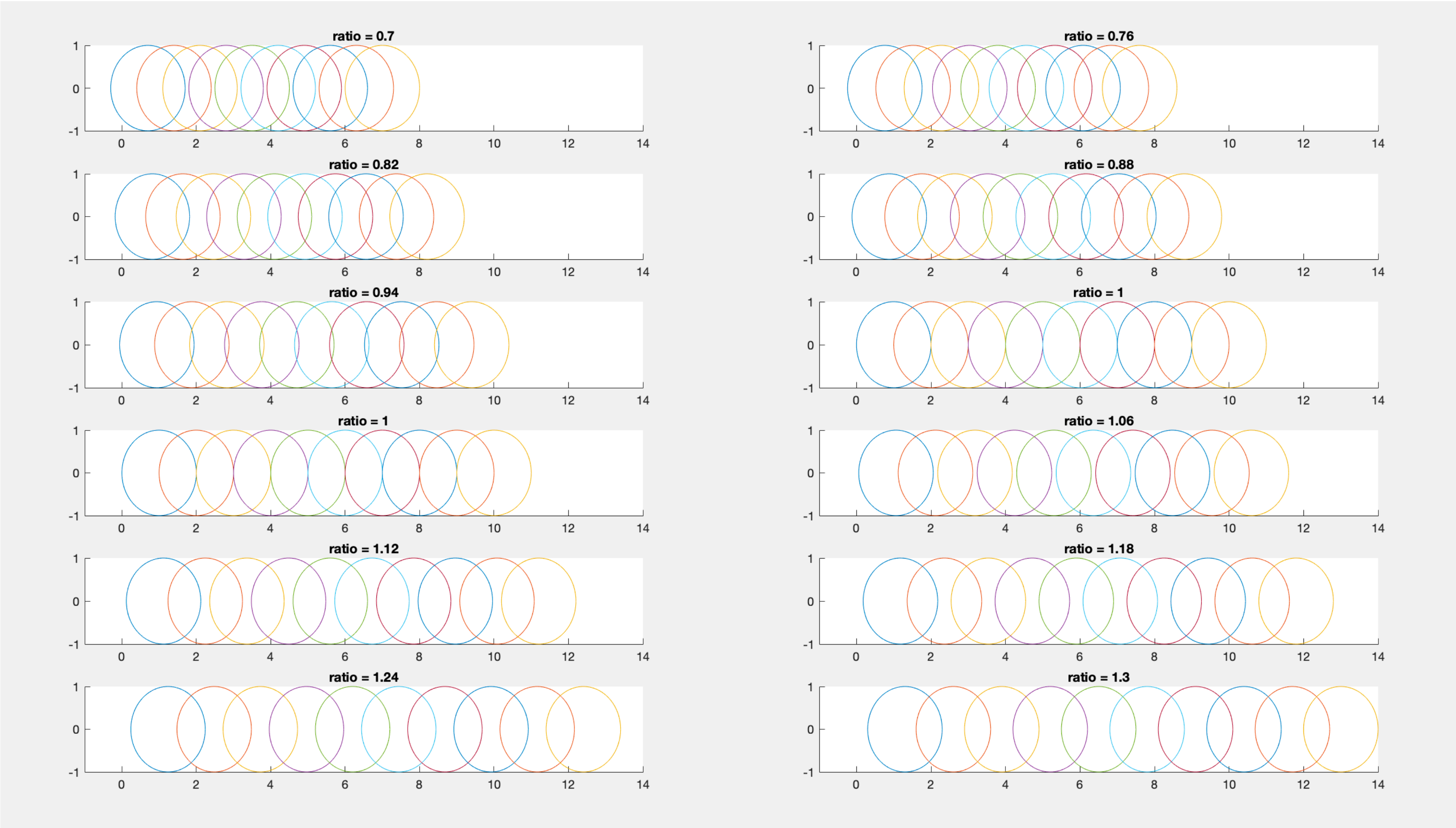
IV. Feature 2: Slack Variable for Intensional Violation

$$\boldsymbol{n}^T \Delta \boldsymbol{x} \geq -\boldsymbol{n}^T (\boldsymbol{x} - \boldsymbol{p}) \quad \longrightarrow \quad \begin{aligned} \boldsymbol{n}^T \Delta \boldsymbol{x} &= -\boldsymbol{n}^T (\boldsymbol{x} - \boldsymbol{p}) + r + s \\ s &\geq 0 \end{aligned}$$

s is the slack variable, this allows surgeons to violate some of the constraints to an extent measured by the slack variable in order to perform surgical operation.

IV. Feature 3: Whole Shaft Modeling

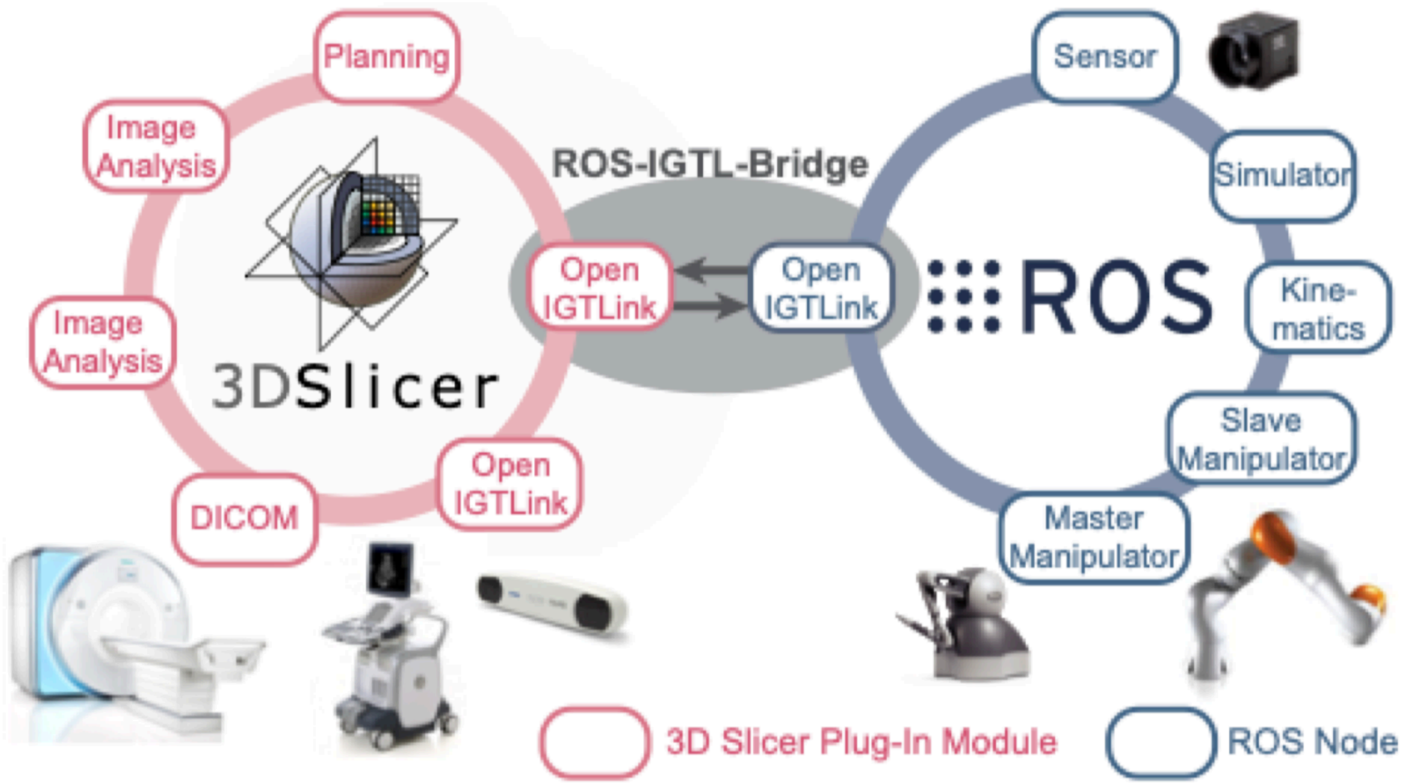
- Previously, we've modeled the tool tip as a point. However, practically, the whole tool shaft should all be considered in the motion planning algorithm.
- Based on the current point-based algorithm, we first extend a point model to a sphere model and then propose a sampling based method by which the current algorithm can be extended to the scope of the whole tool shaft. Generated some points evenly distributed in the region of the tool shaft and generate a sphere at each sample points.
- Added 3 rotation variables to the state x, extended from 3-Dof to 6-Dof.
- Adjusted the Jacobian matrix accordingly.
- Modified the construction of the constraint matrix, enabling it to take all the active constraints of all the sample points, with regard to 6 state varaibles.


ratio = 0.7
ratio = 0.76
ratio = 0.82
ratio = 0.88
ratio = 0.94
ratio = 1
ratio = 1
ratio = 1.06
ratio = 1.12
ratio = 1.18
ratio = 1.24
ratio = 1.3

IV. Feature 4. Fast Autonomous Retraction

- Autonomous backward motion tracking forward path
  - To retrieve tools after operation with a known safe path, minimizing the retraction time
- Higher control frequency during reversal for speed boosting
  - Increase average speed
  - Allow higher control frequency at 10k Hz vs. base operation at 1k Hz
- Partition of original trajectory & speed averaging
  - For each incremental displacement, equally divide it into multiple segments such that the new segments of the whole trajectory have similar length
  - Commanding through the new trajectory backward with consistent higher frequency results in a higher speed
- Performance metric
  - Error between previous trajectory and actuated trajectory
  - Smoothness of the velocity

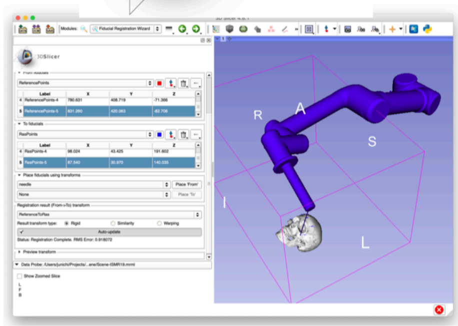
V. Simulation Environment



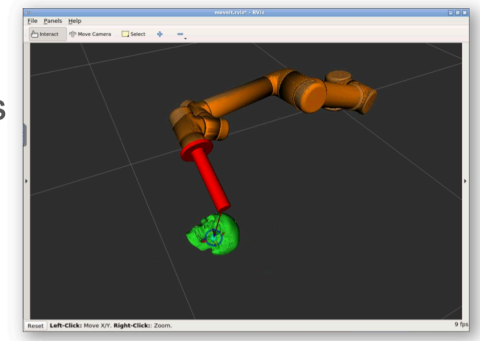
VI. Architecure

FRONT

3DSlicer



ROS



Target/entry points

Positions of links

3D medical image (MRI)
- Surgical planning
- Procedure monitoring

Robot Arm
- Kinematics
- Path planning

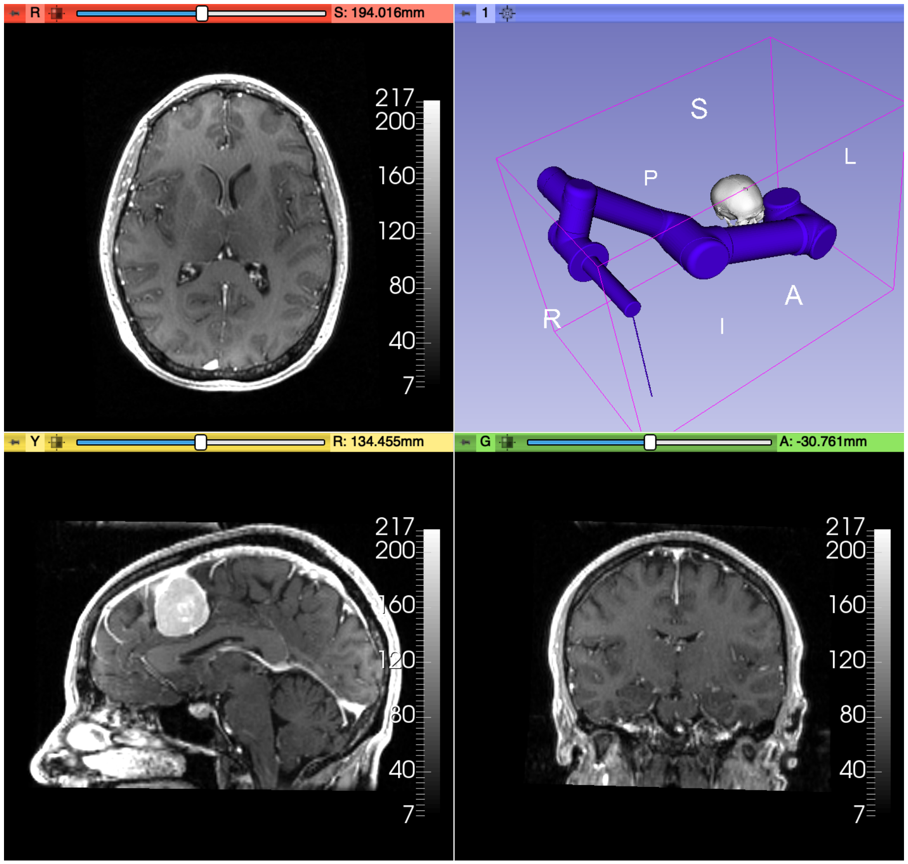
BACK

CISST ⇄ SAW

Data Structure

Surgical Robot Model

VII. Place Holder for the video demo



Timeline, Milestones & Deliverable

- **Step 1: Improve the data loading speed** Mar. 5 ~ Mar. 19 (14 days)
  - Write proposal, give presentation and construct the wiki page.
  - Get familiar with the CISST code base
    (Note: mainly focused on numerical environment, and the Constraint Controller part)
  - **Milestone:** Complete the task of loading mesh stl binary file and add it to the library. (Feature1)
  - **Status:** 100%

- **Step 2**: **Implement the Slack Variable for Constraint Violation** Mar.20 ~ Apr. 5 (14 days)
  - Get familiar with the Galen Robot code base
  - Adding slack variable of soft constraints to the optimal controller.
    (Note: This is also the minimum deliverable)
  - **Milestone:** Pass compilation of the integrated Galen Robot cotroller. (Feature2)
  - **Status:** 100%

Timeline, Milestones & Deliverable

- **Step 3: Test the integrated controller in simulation, using simple robot model** Apr. 6 ~ Apr. 20 (14 days)
  - Test the controller with a simple robot model (eg. UR5 robot arm)
  - Test the controller with 3-Dof translation-only motion.
  - Test the controller with simple geometry obstacle.
  - Test the controller with 3D phantom of patient anatomy. (online data source)
  - Improving the mesh-constraint algorithm with respect to the whole surface (use sphere model) of the end-effector, rather than modeling it with a point. (Note: This is also the expected deliverable)
  - Test the controller with 6-Dof motion.
  - Milestone: A video to demonstrate a simple robot can actually perform the above tasks in simulation. (Feature3)
  - Status: 90%

Timeline, Milestones & Deliverable

- **Step 4: Test the integrated controller in simulation, using Galen robot model** Apr. 21 ~ Apr. 30 (10 days)

  - Test the controller with a simple robot model (eg. UR5 robot arm)
  - Test the controller with 3-Dof translation-only motion.
  - Test the controller with simple geometry obstacle.
  - Test the controller with 3D phantom of patient anatomy. (online data source)
  - Improving the mesh-constraint algorithm with respect to the whole surface (use sphere model) of the end-effector, rather than modeling it with a point. (Note: This is also the maximum deliverable)
  - Test the controller with 6-Dof motion.
  - Milestone: A video to demonstrate the Galen robot can actually perform the above tasks in simulation.
  - Status: 0%, due to the failed dependency on Galen robot visualization model.

Acknowledgement

- Zhaoshuo Li et al. Anatomical Mesh-Based Virtual Fixtures for Surgical Robots (unpublished by Mar. 2020)
- Funda, J., Taylor, R. H., Eldridge, B., Gomory, S., & Gruben, K. G. (1996). Constrained Cartesian Motion Control for Teleoperated Surgical Robots. Robotics, 12(3).
- Xia, T., Kapoor, A., Kazanzides, P., & Taylor, R. (2011). A constrained optimization approach to virtual fixtures for multi-robot collaborative teleoperation. IEEE International Conference on Intelligent Robots and Systems, 639–644. https://doi.org/10.1109/IROS.2011.6048816
- Li, M., Ishii, M., & Taylor, R. H. (2007). Spatial Motion Constraints Using Virtual Fixtures Generated by Anatomy. 23(1), 4–19.
- Kapoor, A. (2008). Motion constrained control of robots for dexterous surgical tasks.
- Masotidectomy Video https://www.youtube.com/watch?v=jnonLwxW2Cg
- STL parser http://www.dillonbhuff.com/?p=5
- STL reader https://github.com/sreiter/stl_reader
- Simulation Environment Tutorial https://rosmed.github.io/ismr2019/prerequisite
- CISST-SAW Open Source https://github.com/jhu-cisst/cisst/wiki

Q&A Session

Thank you!