

# A Surgical Planning Toolkit for Robotic Stereotactic Neurosurgery

by

Dhruv Kool Rajamani

A Thesis

Submitted to the Faculty

of the

WORCESTER POLYTECHNIC INSTITUTE

In partial fulfillment of the requirements for the

Degree of Master of Science

in

Robotics Engineering

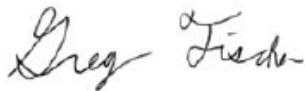
by



---

July 2021

Approved by:



---

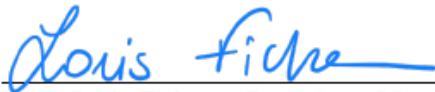
Prof. Gregory S. Fischer, Advisor  
Worcester Polytechnic Institute

Approved by:



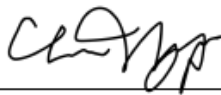
---

Prof. Gregory S. Fischer, Advisor  
Worcester Polytechnic Institute



---

Prof. Loris Fichera, Committee Member  
Worcester Polytechnic Institute



---

Dr. Christopher J. Nycz, Committee Member  
Worcester Polytechnic Institute



## **Abstract**

Robotic Image-guided surgeries (RIGS) have flourished in the past two decades. Advancements in the field have led to a wide adaptation of this technology in many areas of surgery including the tissue biopsy, tumor ablation, and targeted drug delivery. One example is the MRI-guided stereotactic robotic-assisted surgery for conformal brain tumor ablation, where the robot is used to position and orient a thin probe to target a desired region within the brain. Requirements such as the remote center of motion and precise manipulation, impose the use of complex kinematic structures, which result in non-trivial workspaces in these robots. The lack of workspace visualization poses a challenge in selecting valid entry and target points during the surgical planning and navigation stage.

This thesis presents a surgical planning toolkit for an MRI-compatible stereotactic neurosurgery robot. The toolkit aims to improve the accuracy of the surgical intervention and reduce the operative time by augmenting the surgeons field of vision. The tools available to the surgeon help render and augment the robot's reachable workspace over the MRI volume, identify the cranial burr-hole and an optimal Entry Point, and finally render the effective sub (treatment) workspace to assist the surgeon in placing a Target Point within the bounds of the robot's reach.

Finally, the tools presented are validated for accuracy using a variety of experiments. The accuracy of the workspace visualization is validated using phantom studies. The accuracy of semi-automated burr-hole identification and Entry Point selection is validated using a set of experiments. A user study was carried out to identify the deviation between Entry Points placed by an expert neurosurgeon with

and without assistance of the toolkit. Finally, the applicability of the toolkit in a clinical environment was demonstrated through subacute pig studies.

## Acknowledgements

I am grateful to my advisor Professor Gregory Fischer for supporting me throughout the graduate program at WPI. He provided me the agency to explore the field of medical robotics and helped me succeed in all projects I undertook. I thank my committee members Professor Fichera, and Dr. Christopher Nycz for reviewing my thesis on such short notice. I especially thank Dr. Christopher Nycz for co-advising me during my project on the HOPE Hand Exoskeleton and helping me run experiments to validate my thesis.

I admit I would not have reached this far without the assistance of both Farid (Ryan) and Ben who have worked with me on this project. I thank my colleagues - Dr. Adnan Munawar for his sagely wisdom and mentorship while he was still a student at the AIM Lab, Vignesh and Raunak for keeping me company during sleepless nights in the lab, and Dr. Nathalie Majcherczyk for livening up the atmosphere of the lab and bolstering my confidence whenever I would hit a block.

I owe my success to my parents for their unconditional support. My journey to becoming a Roboticist would be nonexistent without your prescience for my passion for robotics which was unbeknownst to me at the time. To my sister Naira, thank you for keeping me grounded and holding the fort for the both of us back home. I appreciate my friends for keeping me sane through this pandemic by joining me on weekly zoom calls, and Avnish for never letting me be alone no matter how far I go.

This work was funded by the National Institutes of Health (NIH) under the National Cancer Insitutiton (NCI) (Grant No. R01 CA166379).

# Contents

<b>1</b>	<b>Introduction</b>	<b>1</b>
1.1	Prior Work . . . . .	4
1.1.1	Current Workflow . . . . .	5
1.1.2	Modified workflow from [33] . . . . .	6
1.2	Problem Formulation and Contributions . . . . .	9
1.2.1	Problem Formulation . . . . .	9
1.2.2	Contributions . . . . .	11
<b>2</b>	<b>Background</b>	<b>13</b>
2.1	Medical Imaging . . . . .	13
2.1.1	Magnetic Resonance Imaging (MRI) . . . . .	15
2.1.1.1	MR: System Design . . . . .	15
2.1.1.2	MR: Basic Physics [49] . . . . .	16
2.1.1.3	MR: Sequences . . . . .	18
2.1.2	Computed Tomography (CT) . . . . .	18
2.2	Technologies . . . . .	21
2.2.1	3D Slicer . . . . .	21
2.2.2	Visualization ToolKit [57] . . . . .	21
2.2.3	Meshlab [58] . . . . .	22

## CONTENTS

2.2.4	Volume Rendering . . . . .	22
2.3	Image Segmentation Techniques . . . . .	23
2.3.1	Watershed Segmentation [61] . . . . .	24
2.3.2	Flood-filling [63] . . . . .	24
2.3.3	Fast-marching [64], [65] . . . . .	25
2.3.4	Deep Learning in Medical Imaging . . . . .	26
2.3.5	NVIDIA Clara [76] . . . . .	26
2.3.5.1	NVIDIA Artificial Intelligence Assisted Annotation [78] . . . . .	27
2.4	NeuroRobot Overview . . . . .	27
2.4.1	Robot Design and Kinematics . . . . .	29
2.4.2	Controlling the NeuroRobot [84] . . . . .	29
2.4.2.1	Hardware . . . . .	29
2.4.2.2	Software . . . . .	31
2.5	Neuroablation . . . . .	31
2.6	Burr-hole Detection using Medical Imaging . . . . .	33
<b>3</b>	<b>Surgical Planning Toolkit</b>	<b>35</b>
3.1	Requirements . . . . .	35
3.1.1	Application Requirements . . . . .	35
3.1.2	Design Requirements . . . . .	36
3.2	Architecture . . . . .	36
3.2.1	Graphical User Interface . . . . .	38
3.2.2	Medical Reality Modeling Language Node . . . . .	41
3.2.3	Logic . . . . .	45
3.2.3.1	Workspace Generation . . . . .	46
3.2.3.2	Workspace Visualization . . . . .	48

# CONTENTS

3.2.3.3	Entry Point Identification . . . . .	51
<b>4</b>	<b>Experiments</b>	<b>58</b>
4.1	Validating generated workspaces . . . . .	58
4.1.1	Procedure . . . . .	58
4.1.2	Results and Discussion . . . . .	61
4.2	Validating EP Identification . . . . .	61
4.2.1	Procedure . . . . .	62
4.2.2	Results and Discussion . . . . .	66
4.3	User study for EP identification . . . . .	70
4.3.1	Procedure . . . . .	71
4.3.2	Results and Discussion . . . . .	71
4.4	Pig trials with modified workflow . . . . .	73
4.4.1	Modified Workflow . . . . .	75
<b>5</b>	<b>Conclusion</b>	<b>77</b>
5.1	Key Contributions . . . . .	77
5.1.1	Structured surgical workflow design . . . . .	77
5.1.2	Intuitive toolkit design . . . . .	78
5.1.3	Experimentation and Validation of the toolkit's components . . . . .	79
5.2	Impact of COVID on this work . . . . .	80
5.3	Future Work . . . . .	81
5.3.1	Feature: Entry Point Region Identification . . . . .	81
5.3.2	Feature: Enabling communication between the Robot Control Software and Surgical Planning Toolkit . . . . .	83
5.3.3	Feature: Motion Planning for collision avoidance . . . . .	83
5.3.4	Feature: Surgical Planning during insertion . . . . .	84

## CONTENTS

5.3.5	Modifications to the current toolkit . . . . .	85
5.3.5.1	Burr-hole segmentation . . . . .	85
5.3.5.2	Workspace Visualiztion . . . . .	86

# List of Figures

1.1	A traditional craniotomy taking place without image guided assistance. [6]	2
1.2	The stryker CranialMap system.	3
1.3	Typical surgical workflow for conformal brain tumor ablation using the NeuroRobot.	5
1.4	The robot when placed on the scanner bed before the registration sequence takes place.	7
1.5	A flowchart of the current surgical workflow also mentioned in figure 1.3.	8
1.6	A rendering showing the deviation of the probe when it collides with the edge of the burr-hole.	11
2.1	MR Scanner schematic diagram.	16
2.2	A schematic of a CT Scanner.[53]	19
2.3	CT image of a burr-hole drilled in a sheep head.	20
2.4	The 3D Slicer Application as seen on startup.	20
2.5	An interactive window to set the transfer function to render a volume.	23
2.6	Input and output for watershed segmentation using 3D Slicer, taken from [62].	24



## LIST OF FIGURES

2.7	The result of flood-filling by clicking on the tumor. Taken from [62]. . .	25
2.8	Input and output for fast-marching segmentation using 3D Slicer, taken from [62]. . . . .	25
2.9	A Leksell stereotactic frame [10], (Elekta Instruments, Stockholm, Sweden) . . . . .	28
2.10	NeuroRobot side view demonstrating RCM. . . . .	29
2.11	Labelled rendering of the NeuroRobot. . . . .	30
2.12	Robot control box. [84] . . . . .	30
2.13	The robot control interface. [84] . . . . .	31
2.14	The NBTU probe developed by Acoustic MedSystems™. [84] . . . . .	32
2.15	Burr-hole as seen in an MRI image of a pig. . . . .	33
3.1	Architecture of the Surgical Planning Toolkit. . . . .	37
3.2	Labelled image of the GUI for the surgical planning toolkit. . . . .	40
3.3	Popups created by the GUI and Logic to direct the surgeon’s attention to an important step. . . . .	41
3.4	NeuroRobot’s workspaces generated by the Surgical Planning toolkit. . . . .	46
3.5	Mesh generation using the MeshLab server. . . . .	48
3.6	Qualitative comparison of burr-hole segmentation techniques. . . . .	54
4.1	Phantom targeting test using the surgical planning toolkit. . . . .	60
4.2	Preparing the sheep head for the study. . . . .	64
4.3	Setting up the NeuroRobot and Sheep head for the procedure. . . . .	65
4.4	Placing Slicer markers where the MR/CT marker centers are visible. . . . .	66
4.5	3D rendered volume of the registered MR-CT scans. . . . .	67
4.6	RAS planes of registered CT and MR scans along with 3D rendered view of the registered volumes. . . . .	67

## LIST OF FIGURES

4.7	A plot of all the magnitude errors for each trial. . . . .	68
4.8	A segmented burr-hole from the first sheep head which is used as ground-truth. . . . .	69
4.9	Assisted segmentation of the burr-hole region. a) 8 markups placed inside the burr-hole region to annotate the extremum. b) Result of the assisted segmentation using the surgical planning toolkit, along with EP identification. . . . .	73
4.10	The proposed workflow as performed using the surgical planning toolkit.	75
5.1	A rendering of an ROI showing the cross-sectional area of the drill bit.	82
5.2	A 3D CAD imported mesh of the drill bit aligned with the insertion angle of the burr-hole. . . . .	82
5.3	The NeuroRobot loaded in RViz with markers for waypoints generated by the motion planning service in Bove's dissertation [33] . . . .	84
5.4	Surgical planning with collision avoidance using the PyDBS extension in Slicer [20] . . . . .	85

# List of Tables

4.1	Ground-truth, and Assisted estimate locations of burr-hole centers in RAS coordinates for each sheep subject. <b>G</b> - Ground-truth from CT Image, <b>A</b> - Assisted estimate from toolkit on MR image, <b>E</b> - Error (deviation) between ground-truth and assisted estimate using the toolkit, <b>ME</b> - Magnitude Error in mm, <b>S1</b> - Sheep 1, <b>S2</b> - Sheep 2, <b>S3</b> - Sheep 3, <b>Sub</b> - Subject, <b>Pos</b> - Burr-hole position, <b>L</b> - Left, <b>R</b> - Right. . . . .	68
4.2	Ground-truth, and Assisted estimate segments of burr-holes in $mm^3$ for each sheep subject. <b>E</b> - Error (deviation) between ground-truth and assisted estimate, <b>S1</b> - Sheep 1, <b>S2</b> - Sheep 2, <b>S3</b> - Sheep 3, <b>Sub</b> - Subject, <b>Pos</b> - Burr-hole, <b>L</b> - left, <b>R</b> - right. . . . .	70
4.3	Comparing the absolute difference between EP placed by the expert with and without assistance for all successful trials. . . . .	71
4.4	Comparing the absolute error between assisted EP identification over two trials on the same burr-hole. The first trial was unsatisfactory to the surgeon, the second trial was more than satisfactory. . . . .	72

# Listings

3.1	Burr-hole parameters class specification . . . . .	42
3.2	vtkMRMLSurgicalPlanningNode class specification. This listing fo- cuses on discussing the event handling by the MRML Node. . . . .	43
3.3	XML script which contains all the steps for applying filters to the desired mesh . . . . .	49
3.4	Watershed segmentation implemented in ITK. . . . .	52
3.5	Flood-filling segmentation implemented in VTK and Slicer. . . . .	52
3.6	Fast-marching segmentation implemented in ITK. . . . .	53
3.7	Island removal by searching for the largest segment using VTK. . . .	56

# Acronyms

**AI** Artificial Intelligence

**AIAA** AI Assisted Annotation

**API** Application Programming Interface

**BRAVO** Brain Volume Sequence

**CT** Computed Tomography

**DOF** Degrees Of Freedom

**EP** Entry Point

**EPW** Entry Point Workspace

**FK** Forward Kinematics

**FLAIR** Fluid Attenuated Inversion Recovery

**GUI** Graphical User Interface

**IGI** Image Guided Interventions

**IK** Inverse Kinematics

**MRI** Magnetic Resonance Imaging

**MRML** Medical Reality Markup Language

**MVC** Model-View-Controller

**NBTU** Needle-Based Therapeutic Ultrasound

**NIH** National Institutes of Health

**RAS** Right Anterior Superior coordinate system

**RGBA** Red Green Blue Alpha color scheme

**SNR** Signal-to-Noise Ratio

**SW** Sub-Workspace

**T1w** T1-Weighted

**TP** Target Point

**TRW** Total Reachable Workspace

**VTK** Visualization Toolkit

# Chapter 1

## Introduction

Progress towards image-guided robot-assisted surgery, not only requires development of compatible hardware, but also an interactive software framework to provide pre-operative and intra-operative surgical planning capabilities to the physicians. Some benefits of surgical planning are providing physicians with useful information regarding the feasible tool and robot trajectories [1], selection of the optimal entry and target points [2], segmentation and reconstruction of various anatomical structures [3], and the ability to define virtual fixtures that act like no-fly-zones to avoid collision of the robot with crucial anatomy [4]. Moreover, surgical planning can streamline the surgical workflow by providing a more intuitive understanding of the robot's motion and its workspace and helps physicians achieve higher operational accuracy and increased efficiency [5]. Figure 1.1 shows a traditional open neurosurgical craniotomy without image guided assistance.

Several Image Guided Interventions (IGI) are currently in use in clinical practice for neurological and neurosurgical applications. In general, neurosurgical IGI can be either open, minimally invasive, or non-invasive. Examples of open neurosurgical interventions include the use of computer-based neuronavigation applications



Figure 1.1: A traditional craniotomy taking place without image guided assistance. [6]

(Stryker CranialMap[7] also seen in figure 1.2, Medtronic Stealth[8], and Brainlab[9]) for the excision of brain tumors. Advantages of neuronavigation include the ability to plan smaller craniotomies and assess cranial anatomy and extent of resection of tumors in real-time. Examples of non-invasive IGI include stereotactic radiosurgery (SRS) [10] for the treatment of brain metastases and transcranial MRI-guided focused ultrasounds (tcMRgFUS) [11] for lesioning. Examples of minimally invasive neurosurgical IGI include laser interstitial thermal therapy (LITT) [12], [13], stereotactic biopsy [14], [15], and placement of stereotactic electroencephalography (sEEG) depth electrodes [16] or radiofrequency ablation (RFA) probes [17]. There are many benefits of the less invasive procedures possible with IGI including smaller incisions, lower infection rates, decreased morbidity and mortality, and enhanced recovery [18], [19].





Figure 1.2: The stryker CranialMap system which offers image guided assistance during a neurosurgery through an intra-operative navigation camera and overlays images on preoperative scans. [7]

Use of robots in image-guided stereotactic neurosurgery procedures stems from their ability to provide precise motion of the end effector and actively monitoring and correcting the tool path under the image guidance[20]. These capabilities give these systems a great advantage over the traditional manual stereotactic frames by increasing accuracy and decreasing operation time. A recent study shows a large

adaptation of these robots for a wide range of *in vivo* neurosurgical procedures[21], including functional neurosurgery[22], sEEG placement for epilepsy[23], and deep brain stimulation lead placement for Parkinson’s disease[24]. Additionally, these robots are used for therapeutic interventions mostly to destroy pathological tissue such as brain tumors[25] and seizure foci[26]. In these procedures, access to the brain is achieved through a roughly 1 cm or smaller opening on the cranium called a burr-hole, which in comparison to the more traditional open-surgery approach, dramatically decreases the invasiveness of the surgery and recovery time. The authors in [27] conclude that these platforms have shown great promise in terms of increased safety and accuracy, cost-effectiveness, and compactness and conclude that the acceptance of these surgeries is becoming more commonplace.

## 1.1 Prior Work

We have previously designed and developed an MRI-compatible stereotactic neurosurgery robot (NeuroRobot) for in-bore closed-loop conformal ablation of brain tumors using an interstitial ultrasound probe to deliver needle-based therapeutic ultrasound (NBTU) [29]. The robot provides positioning and alignment of the ablation probe given desired Entry Point (EP) and Target Point (TP), as well as the probe rotation to enable conformal ablation of tumors using directional probes[30].

Each conformal tumor ablation procedure using the NeuroRobot follows a systemic pre-operative surgical planning approach to ensure safe and accurate operation to maximize the surgical outcome. A typical workflow for conformal ablation using the NeuroRobot is shown in figure1.3. The feasibility and workflow of MR-guided robotically assisted delivery of NBTU has been previously demonstrated by our group[31], [32] in large animal studies in preparation for human use.

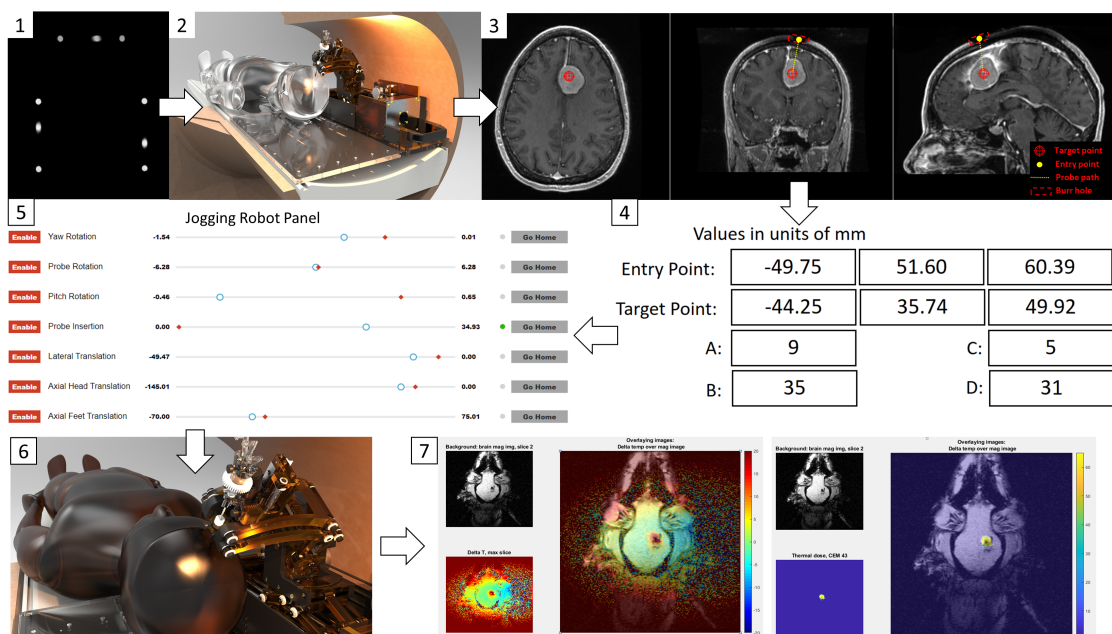


Figure 1.3: Typical surgical workflow for conformal brain tumor ablation using the NeuroRobot. 1) A registration scan is taken to register the robot with respect to the imager’s frame. 2) Subject is placed and secured on the MRI bed. 3) A series of the pre-operative scans are acquired and visualized on the 3D Slicer software. The Physician locates the burr-hole on the MRI image and selects the desired EP and TP using fiducial markups. 4) EP and TP are sent to the robot’s inverse kinematics solver through OpenIGTLink[28] to evaluate reachability of those points. 5) If EP and TP are validated, the software returns desired joint values. In case the points are invalid, the physician is asked to select another EP and TP and repeat step 4. 6) The robot is moved into the targetting configuration and the probe is inserted into the brain. 7) Intra-operative scans are acquired to monitor the applied thermal dosage, and are followed by post-ablation scans to validate the accuracy and procedure results.

### 1.1.1 Current Workflow

As seen in figures 1.3 and 1.5, a typical experiment is carried out by following these steps:

- 1) The subject (pig) is carried into the operating room and a unilateral or bilateral burr-hole is drilled.
- 2) The robot is attached to the scanner bed (figure 1.4) and a registration scan

- (T1w BRAVO sequence) is performed to register the robot with respect to the imaging frame. (figure 1.3(1))
- 3) The subject (pig) is carried to the MRI suite and placed on the scanner bed. (figure 1.3(2))
  - 4) A series of T1 scans are acquired and loaded into Acoustic MedSystems Theravision after which the operating surgeon identifies the burr-hole from the image and places an Entry Point (EP) marker at the holes center, and a Target Point (TP) marker at the tumor/target to ablate. (figure 1.3(3))
  - 5) The EP and TP are pushed to the Robot's controller using OpenIGTLink[28] to compute the Inverse Kinematics (IK) to reach the target point. (figure 1.3(4))
  - 6) If the EP and TP prove to be reachable, the robot's control software returns the joint values to reach these points, otherwise the software alerts the physician to identify new points for the treatment. (figure 1.3(5))
  - 7) The robot is moved to the targeting configuration and the probe is inserted into the brain. (figure 1.3(6))
  - 8) Intra-operative scans are acquired to monitor the applied thermal dosage by the ultrasonic ablation probe. Post ablation scans are taken to validate the accuracy of the procedure. (figure 1.3(7))

### 1.1.2 Modified workflow from [33]

Bove proposed a workflow for the NeuroRobot to visualize the reachable workspace of the Neurorobot, and demonstrated a motion planning algorithm to avoid collisions



Figure 1.4: The robot when placed on the scanner bed before the registration sequence takes place.

while navigating the robot *in-situ* surgery. To achieve this, he designed a ROS package and visualized the reachable workspace in 3D Slicer[34],[35], [36]. This showed potential to reduce the number of times the robot needs to be moved when the EP and TP are not reachable, augmenting step 4. from the current workflow. While this research was a significant contribution to the Neurorobot project it was limited by the software dependencies brought along with ROS and its inefficiency which caused long setup time and slow computation of the reachable workspace. These shortcomings outweighed the benefits and prevented it from being used in a clinical setting.

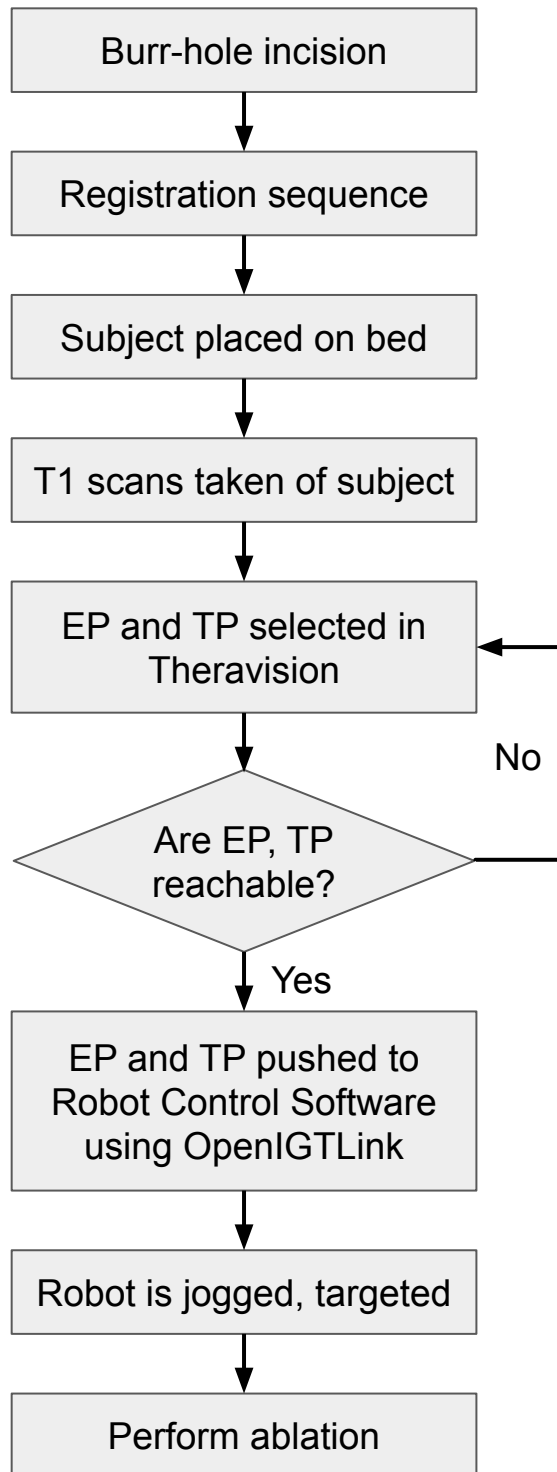


Figure 1.5: A flowchart of the current surgical workflow also mentioned in figure 1.3.

## 1.2 Problem Formulation and Contributions

### 1.2.1 Problem Formulation

Common problems that arise during neuroablation procedures performed by the NeuroRobot are: **1.** Long pre-operative setup time which arise when the burr-hole and/or treatment locations are beyond the reach of the robot in its current configuration, and **2.** uncertainty in EP selection due to loss of visual information of the cranial burr-hole.

1. *Long pre-operative setup time:* The robot is placed on the custom bed and a registration scan is performed before placing the subject on the MRI bed (figure 1.3(1) ). If the desired EP and/or TP lie outside the reachable workspace of the robot, the subject needs to be re-positioned and the scan needs to be performed again. The current method of re-positioning the subject is both experiential and limited by the technician's ability to visualize the robot's workspace during subject repositioning. Setup times can vary based on the technician's experience with prior trials and the size of the subject's head. This is a common occurrence during clinical trials and causes the lead time to setup the robot to increase multi-fold. The root cause of this problem can be traced back to an inability to visualize the robot's workspace while setting up the robot.

This toolkit attempts to overcome this problem by generating the appropriate workspaces and augmenting them on the MRI image. This informs the surgeon whether the EP and TP are valid and provides the technician with an additional modality while re-positioning the subject with respect to the robot.

2. *Uncertainty in selecting an EP:* The EP is the location on the burr-hole where

the probe enters the skull, and an optimal EP allows the largest sub-workspace to be generated and lies equidistant from the edges of the burr-hole. Placing an EP closer to the edge increases the chances of obstruction with the cranium and reduces the effective sub-workspace due to the interference. The current workflow faces uncertainty in the accuracy of identifying the EP with respect to the burr-hole's edges causing the probe to be obstructed by the edges of the burr-hole (figure 1.6). This also causes a deviation in the probe's effective treatment location from the desired treatment location. This is partly due to the oblique orientation of the burr-hole with respect to all three orthogonal image planes ( axial, coronal, and sagittal) which creates an oval representation of a circular burr-hole. Additionally, the visualization of bone in an MRI image is sub-optimal due to low water content in those anatomies since the operating principle of an MRI scanner is to acquire Hydrogen proton-based signals. This adds uncertainty to accurate identification of the burr-hole's edges.

The toolkit also offers an AI based assisted segmentation of the burr-hole to identify and visualize the burr-hole and estimate an optimal EP. The estimated EP is evaluated by the surgeon and can be re-positioned if its unsatisfactory. While the automatic estimation of the EP is an added benefit, the visualization of the segmented burr-hole in all imaging planes is of utmost importance as it adds another modality of information for the surgeon to consider.



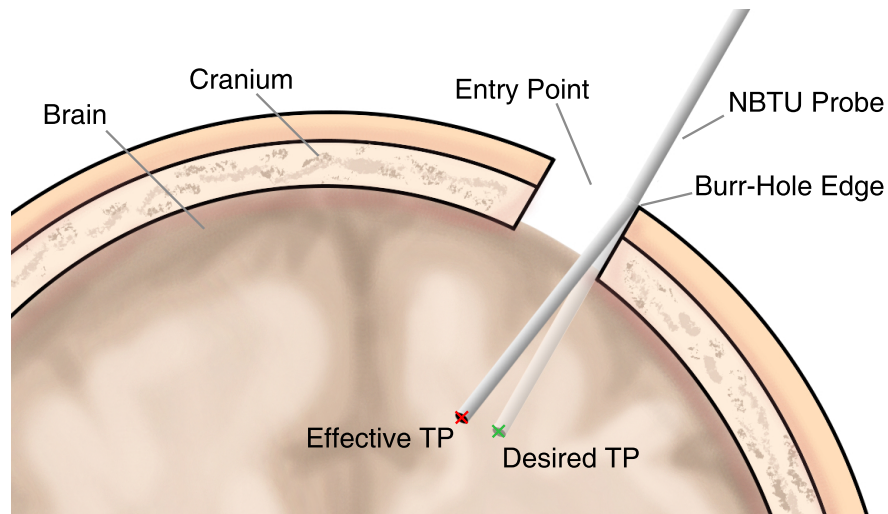


Figure 1.6: A rendering showing the deviation of the probe when it collides with the edge of the burr-hole.

### 1.2.2 Contributions

This work builds upon prior research on surgical planning for the neurosurgery robot by designing a surgical planning toolkit for pre-operative surgeries. The toolkit assists the operating surgeon during the procedure by augmenting the standard MR images with additional visual information, which in turn increases the accuracy of probe insertion and also reduces the pre-operative setup time. Specifically, the contributions have been listed below:

- A surgical planning toolkit for robotic neurosurgery
- Generate reachable, entry-point, and sub (treatment) workspaces
- Assisted segmentation of cranial burr-holes
- Automated Entry Point (EP) identification
- Demonstration during clinical experiments

These contributions have also been tested and validated to reduce the risk of errors as the application for this work is clinical neurosurgery. The following list identifies the tests and validation undertaken to support this work.

- Validation of generated workspace using phantom targeting
- Qualitative comparison of segmentation techniques
- Validation of EP identification
- User study to identify effect of assisted EP identification

In addition, this toolkit combines all the required tools into one unified module for 3D Slicer, eliminating the need to go back and forth from the planning and visualizing software. This could potentially reduce the cognitive load on the surgeon as well as ensure consolidation of information, and also reduce the possibility of errors that could arise with a multi-application software.

This work extensively uses open-source libraries for visualization, segmentation, and a Graphical User Interface (GUI). While the tools used to design this surgical planning toolkit are not novel, there is novelty in the method of assisted burr-hole detection from MR images and the amalgamation of these tools to create a surgical planning workflow. A majority of this work is currently under review [37].

# Chapter 2

## Background

This chapter covers the concepts behind the various tools that have been employed in this project, a concise overview of the NeuroRobot project, and a brief on a typical neuroablation procedure.

### 2.1 Medical Imaging

In 1895, Prof. Willhelm Rontgen revolutionized medicine when he accidentally discovered x-rays [38] and pursued the idea of obtaining images by projecting the rays onto a photosensitive material, creating the field of medical imaging. Today, the field of medical imaging has substantially grown with multiple modalities of imaging techniques being employed. The five major techniques being used are **1)** Magnetic Resonance Imaging (MRI) [39], [40], **2)** Computed Tomography (CT) [41], [42], **3)** Ultrasound Imaging (USI) [43], **4)** Nuclear Imaging like Positron-Emission Tomography (PET)[44], and **5)** X-Ray imaging [38], [45], [46]. Since this project uses both MRI and CT imaging techniques a detailed overview will be provided below along with a shorter description of the other methods presented.

- 1) **Magnetizing Resonance Imaging (MRI)**[39], [40]: MRI is a medical imaging technique that acquire high-resolution cross-sectional images of proton dense objects. There are multiple types of MRI sequencing techniques, and MRI sequences are ideal for imaging soft tissues, blood vessels, etc and can provide critical information to help identify diseases like tumors, aneurysms, multiple sclerosis, pericarditis, and a lot more.
- 2) **Computed Tomography (CT)** [41], [42]: CT is another type of medical imaging technique that takes multiple X-Ray like tomographic images and fuses the cross-sectional slices of the object to generate a Volume. CT scans are very useful in imaging soft and hard tissue and is often the preferred imaging technique over MRI as its cheaper and allows people with magnetic implants to get scans as well. CT scans provide critical information to diagnose fractures, tumors, etc.
- 3) **Ultrasound Imaging (USI)**[43]: USI uses high frequency sound waves in the order of MHz to visualize organs (anatomical USI) or vessels (functional USI). USI is widely used as it doesn't use ionizing radiation and hence is considered safer over X-Rays and other Nuclear Imaging techniques. Its commonly used to view the ovary and uterus during pregnancies, help perform ultrasound guided surgical interventions, evaluate metabolic bone disease, etc.
- 4) **Positron-Emission Tomography (PET)**[44]: PET scans detect oxygen, blood flow and metabolic activity of tissues on a cellular level. Radioactive trackers are consumed by the patient before the imaging procedure which light up tissues that have a higher affinity for these trackers during the scan. PET scans are often used to diagnose cancerous tumors (as they have higher metabolism than the surrounding tissue usually), heart and brain diseases.

- 5) **X-Ray Imaging**[38], [45], [46]: X-Ray imaging is one of the most common and widely used imaging technique which is used to identify bone, some tumors, etc as they absorb radiation more than soft tissue surrounding it. They are useful for detecting fractures and are the most common medical imaging tool available.

### 2.1.1 Magnetic Resonance Imaging (MRI)

As mentioned above, MRI as a diagnostic tool is fairly advanced as it offers high resolution 3D imaging without using ionizing radiation and is fairly safe. This section gives a concise overview on MRI physics and introduces the specific sequences used during this project for neuroablation experiments.

#### 2.1.1.1 MR: System Design

An MRI system (figure 2.1) consists of high power magnet coils for gradients, shim, and radiofrequency sequences, all placed within a single package shaped like a donut with the magnetic fields angled inwards. The system lies within a large copper-lined room which is a Faraday cage, preventing EM radiation from outside the room entering the scanner suite and causing interference. Field strengths of  $1.5T$  and  $3T$  are common for medical imaging scenarios. A workstation outside the MRI suite controls the system within the suite and acquires signals from the MRI machine to convert them into common 3D Medical Imaging formats like Neuroimaging Informatics Technology Initiative (NIFTI), and Digital Imaging and Communications in Medicine (DICOM) [47].

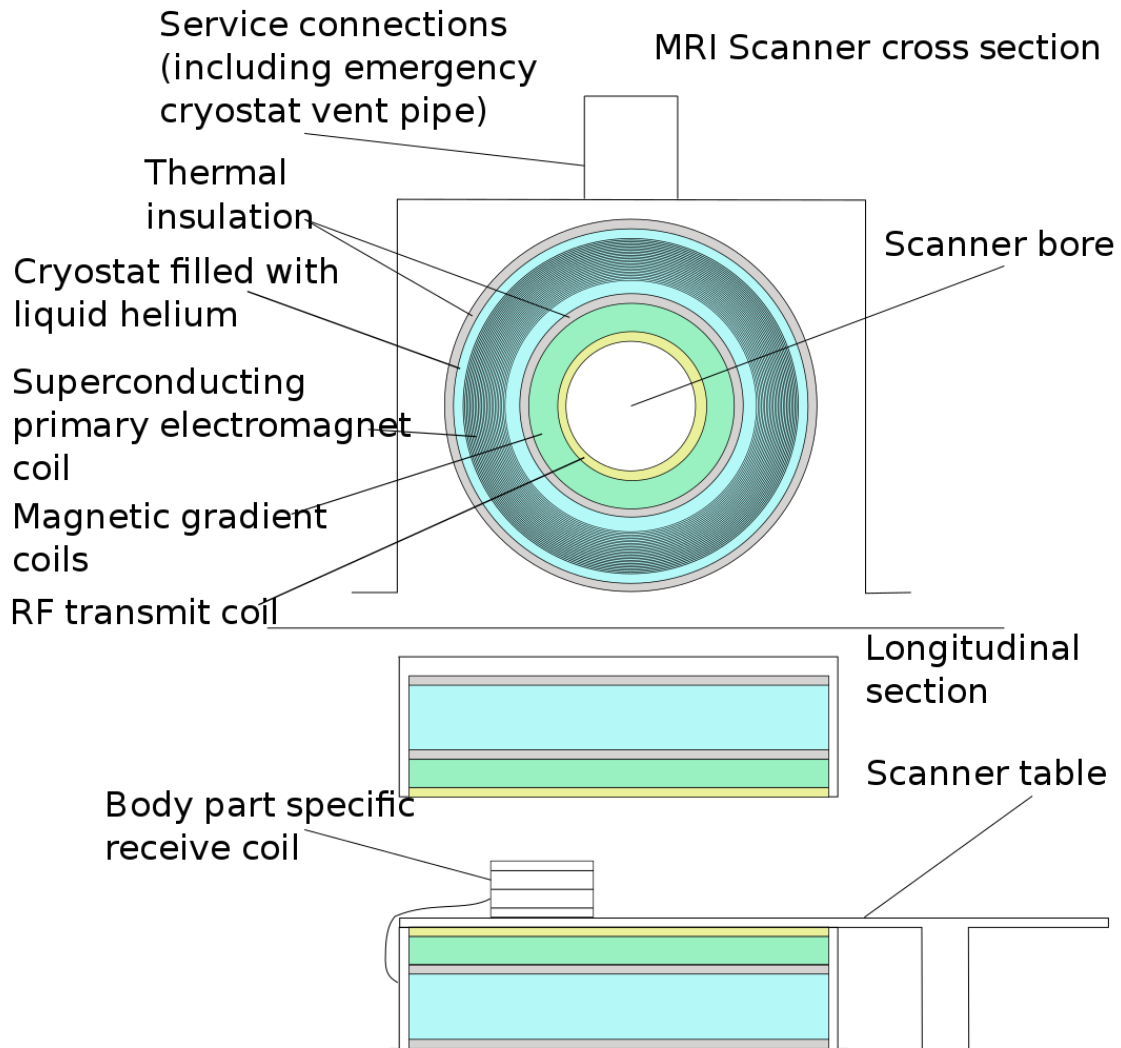


Figure 2.1: Diagram depicting the hardware inside an MR scanner with the different magnet coils, main coils, gradient coils, and RF coils in the order of outermost to innermost. It is common to place target RF coils around the head for better SNR while acquiring MR Images of the brain. From [48]

### 2.1.1.2 MR: Basic Physics [49]

MR leverages the magnetic properties of hydrogen atom that is omnipresent in the human body in the form of water. Hydrogen nuclei consist of a single proton which emits its own magnetic field as its a spinning positively charged particle. When in its natural state, all hydrogen atoms' spin moments are randomly aligned. When

placed in a static magnetic field generated by the magnet coils, all hydrogen atoms align parallel and antiparallel to the applied static magnetic field. This process is also known as precession. Once the protons align parallel to the longitudinal magnetic field they create a strong net directional field  $B_0$  of their own.

RF pulses are now fired with different frequencies to disturb the proton alignment from  $B_0$  by transferring energy to the protons. The frequency of the RF pulses needs to be resonant with the precessional frequency of certain protons, also known as the Larmor frequency. Although certain protons are now misaligned with the field  $B_0$ , a new transverse field spawns from the *in-phase*[49] vibration of the protons. The transverse field is a rotating field spinning at the Larmor frequency and generates an alternating current in a receiver coil which in turn is processed as the MR signal. Once the RF pulses are switched off, the protons return to their original *relaxed*[49] state with a relaxation time  $T2$  for the decaying transverse field, and relaxation time  $T1$  for the returning longitudinal field. When the transverse field starts to decay, protons that were previously *in-phase*[49] with the transverse field start releasing energy to the surroundings at the *tumbling*[49] rate. Free water and hydrogen atoms tend to have relatively long  $T1$  relaxation times, whereas fat tends to have a relatively low  $T1$  relaxation time as the molecular bonds tend to be resonant with the Larmor frequency.  $T2$  relaxation time in comparison doesn't require a decaying field and is influenced by the varying inhomogeneity of different tissue in the body. Cerebro-spinal fluid (CSF) have longer  $T2$  relaxation times as their molecules move very fast and have low inter-molecular interactions as compared to grey matter which has more inter-molecular interactions and hence has a faster  $T2$  relaxation time.

The process of using RF pulses to acquire signals can be time consuming. Gradient Echo (GRE)[50] sequences help reduce the total time to acquire images by

introducing a gradient pulse after an initial RF pulse which results in the protons to go out of phase from the direction of the transverse field at a rapid rate and align with the gradient field. This loss in phase is then reversed by a gradient pulse of equal amplitude in the opposite direction causing the protons to again sync *in-phase* [49] with the transverse field and this process is called GRE. Gradient fields help with localising slices in 3D by generating gradient fields in the direction of a chosen axis. This process takes place in steps known as phase encoding steps.

### 2.1.1.3 MR: Sequences

**FLAIR** FLAIR stands for Fluid Attenuated Inversion Recovery [51] and uses the concept of inversion recovery sequencing changing the visual spectrum of certain tissue by increasing the inversion time. FLAIR attempts to maintain equilibrium with no net transverse field by adjusting the pulse sequences accordingly. The  $T1$  FLAIR sequence is used to register the robot in the imaging frame.

**BRAVO** The 3D Axial Brain Volume [52] sequence is a GE standard sequence that shows exceptional  $T1$  weighted contrast between grey and white matter in brain tissue.

## 2.1.2 Computed Tomography (CT)

CT scans use narrow beam X-rays to capture cross-sectional slices of the subject which are later fused to create a 3D volume of the subject [54]–[56]. As it uses X-ray beams the principle is quite similar with the addition of 3D reconstruction of multiple tomographic slices - The attenuation observed when X-rays pass through tissue help calculate the respective density of the tissue.

Figure 2.2 shows a schematic of a CT scanner. The scanner includes a rotating



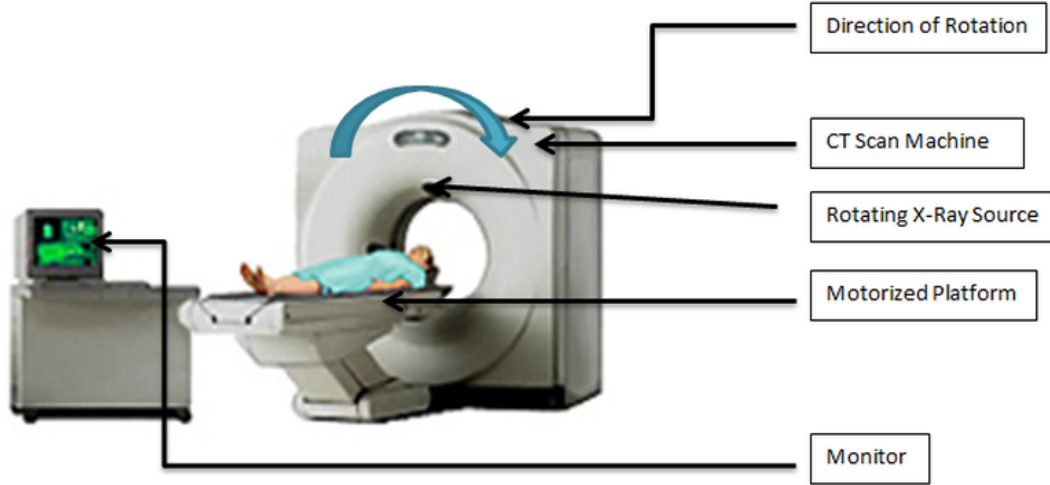


Figure 2.2: A schematic of a CT Scanner.[53]

X-Ray source, a motorized scanner bed, solid-state detectors that help acquire an image and a computer that performs the reconstruction of the slices to make a 3D volume. For the X-ray source, typically, a 60-120 kW generator is used to generate the rays [55], [56]. The rays are acquired by the detector which converts the X-rays into an electrical current. The current is amplified and converted to a digital signal by the workstation and finally an image is reconstructed.

CT images provide high resolution images of all kinds of tissue, especially bone which is hard to acquire in MR images. This project uses CT images (figure 2.3) to identify a ground truth for cranial burr-holes by providing clear distinction between the edge of the burr-hole in the cranium and the gap created where the hole would be.

## CHAPTER 2. BACKGROUND

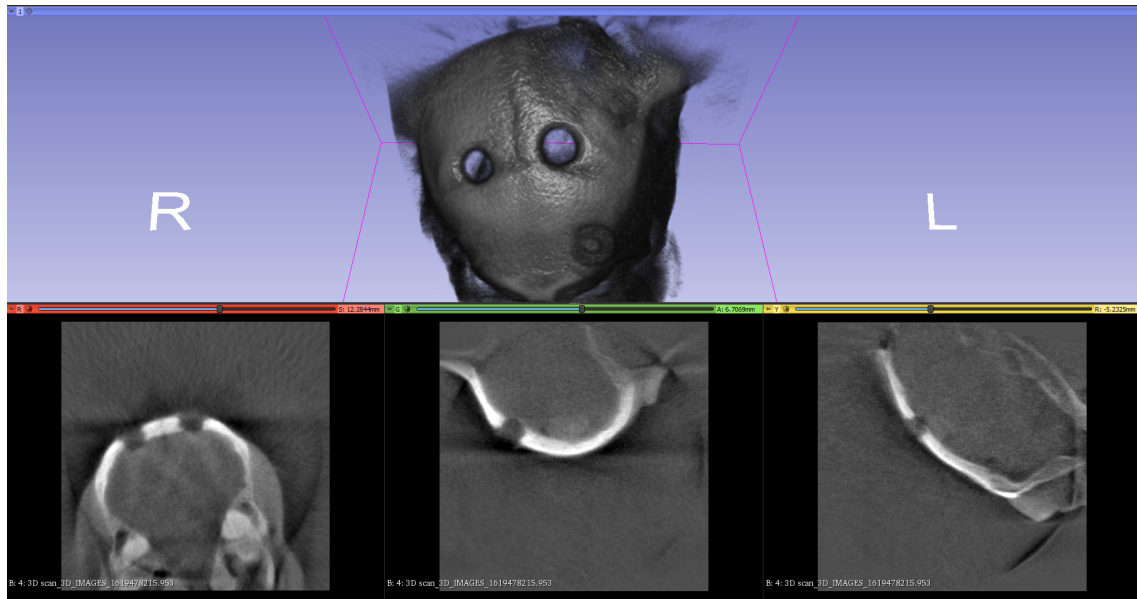


Figure 2.3: CT image of a burr-hole drilled in a sheep head to validate the accuracy of the Entry Point (EP) detection tool.

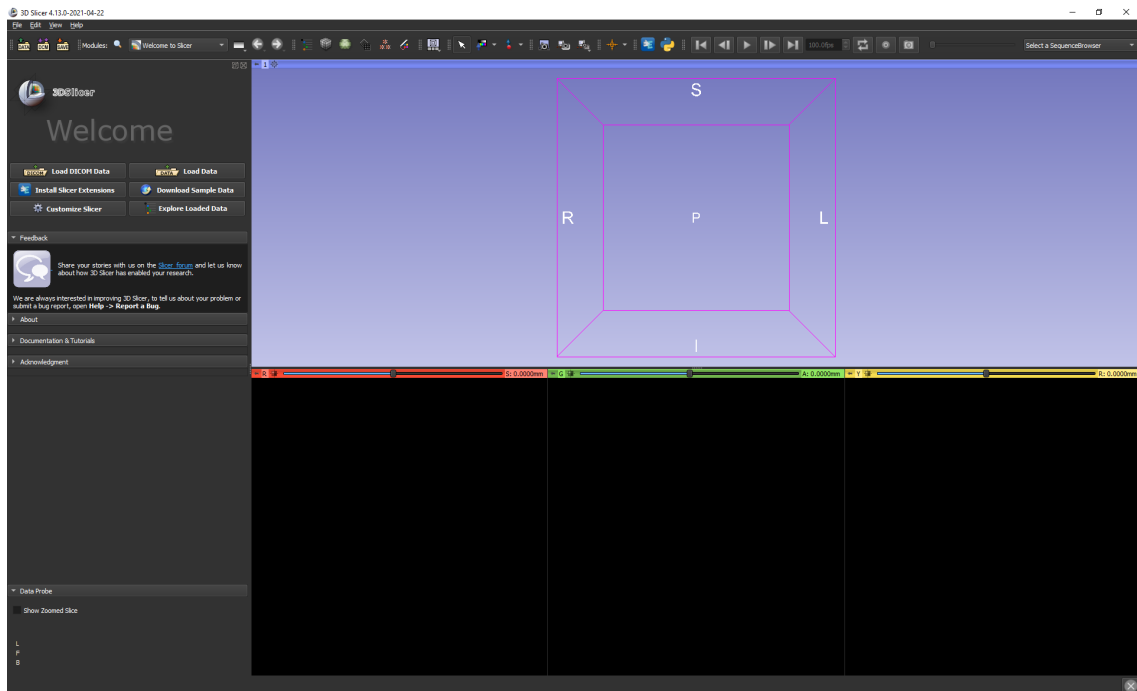


Figure 2.4: The 3D Slicer Application as seen on startup.

## 2.2 Technologies

### 2.2.1 3D Slicer

3D Slicer (figure 2.4) is an open-source platform for the analysis and display of information derived from medical imaging and similar data sets[35]. The software features a broad variety of modules for interactive visualization, image registration, image segmentation, and model-based analysis.

Its extensive developer API provided us with a framework to build our surgical planning toolkit. It follows the Model-View-Controller (MVC) design pattern and communicates between its 3 core components - Graphical User Interface (GUI), Logic, and Medical Reality Modeling Language (MRML) using File Input/Output (I/O), events, and Application Programming Interface (API) calls. Slicer modules wrap this core functionality and offer end-to-end tools for medical image manipulation and analysis.

### 2.2.2 Visualization ToolKit [57]

The Visualization ToolKit (VTK) is an open-source library, implemented in C++ with bindings for Python, and Java based languages, which provides data representation that encompasses point sets (pointclouds (PCL), voxels), polygons, images, volumes, and more [57]. The library provides filters and algorithms for a large number of visualization features like surface and volume reconstruction, pointset downsampling, etc.

### 2.2.3 Meshlab [58]

Meshlab is a 3D mesh processing software which offers a variety of filters, algorithms, and visualization tools to work with pointsets and meshes. The software offers a variety of direct parametric filters to transform a mesh as well as interactive tools to apply filters like smoothing, subsampling, etc, all in real-time. The software uses the Visualization and Computer Graphics (VCG) library extensively which offers highly performant and efficient pointset and mesh processing features. At the time of writing this dissertation, the Meshlab Server, a headless instance of the software that reads an input mesh and process description file and returns the modified mesh, was used to process and reconstruct surfaces of the workspace meshes generated by the extension (Total Reachable Workspace (TRW), Entry Point Workspace (EPW), and Sub-Workspace (SW)).

### 2.2.4 Volume Rendering

All 3D medical imaging techniques capture 2D slices (or tomographs) and reconstruct a 3D image from these slices using volume rendering techniques (figure 2.3 top image in the screen shows the 3D reconstructed view). Since the end product of combining these 2D slices is a 3D volume, each individual block within the 3D volume is called a voxel.

Volume rendering depends on a map of visibility to color which can be achieved using a *transfer function* like a ramp, piece-wise linear function, a table, a spline, or a custom multi-point controlled function [59]. Figure 2.5 shows an interactive window to set the transfer function in 3D Slicer. 3D Slicer also allows the user to load presets for different imaging techniques, sequences, organs, etc. The primary Volume Rendering technique used by 3D Slicer is *volume ray casting* [60] featuring both

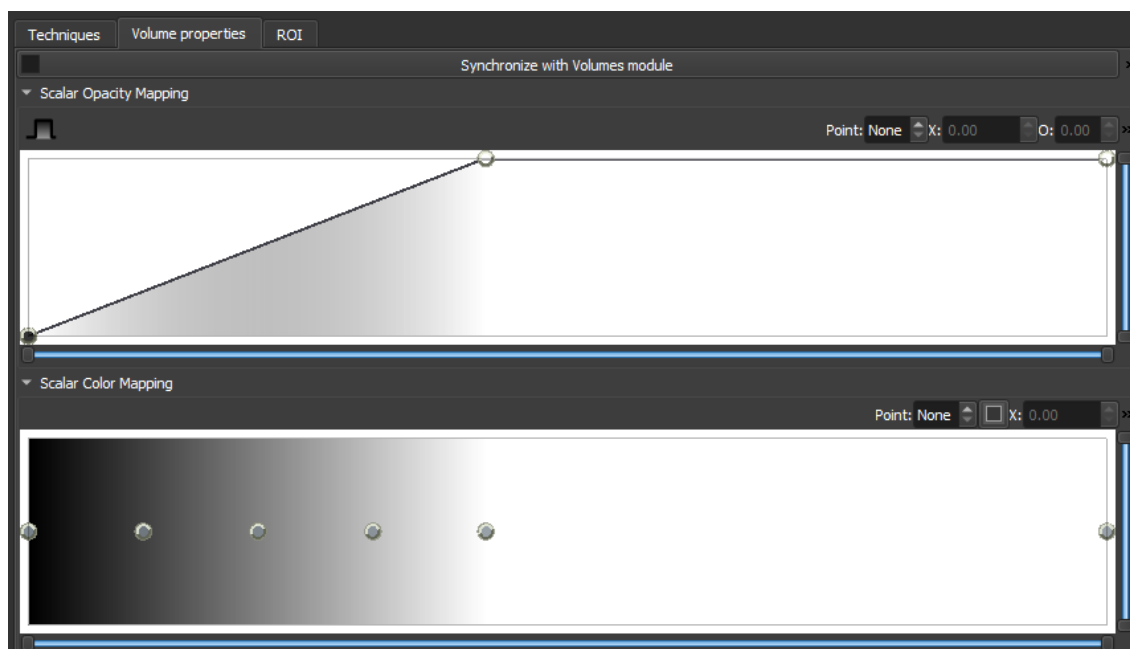


Figure 2.5: An interactive window to set the transfer function to render a volume.

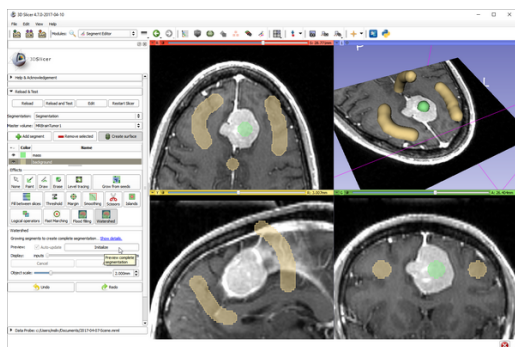
CPU and GPU based variants. Volume ray casting is a high quality image rendering technique and uses the principle of reducing an adapted RGBA interpolation of rays that emanate from the *center of projection* (COP) of a camera and pass through each pixel of the image while being bounded by the region of the volume.

## 2.3 Image Segmentation Techniques

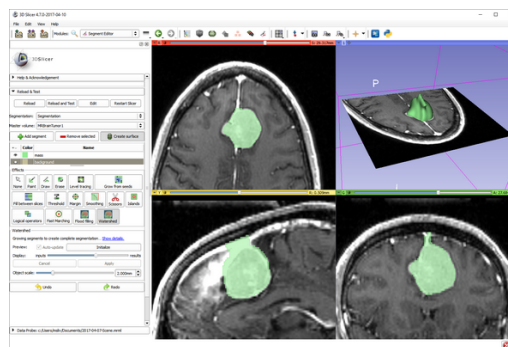
Various image segmentation techniques were explored to assist in segmenting cranial burr-holes to identify the EP. Finally, the Deep Learning (DL) based NVIDIA Artificial Intelligence Assisted Annotation (AIAA) tool was used for this purpose. This section covers the various techniques that were investigated.

### 2.3.1 Watershed Segmentation [61]

Watershed is a region-based segmentation algorithm that reads an image like a topography of peaks and basins. Normally, the pixels' grayscale value is used to establish a peak (white or 0) or basin (black or 255) on an 8-bit scale. Seed points are placed (either manually or automatically) at different regions (figure 2.6a) and mimic independent sources of still water that don't mix. Segmentation takes place as these mock-up sources of water flow across gradients of slopes as morphological transforms. These transforms are limited by boundaries, which mimic a barrage to stop the flow of water from overflowing. A fairly common downside of this technique is over-segmentation which manifests when the medical image is very noisy. The final result is a segmented image as seen in figure 2.6b.



(a) The input image and user annotations to apply watershed segmentation



(b) The result of watershed segmentation

Figure 2.6: Input and output for watershed segmentation using 3D Slicer, taken from [62].

### 2.3.2 Flood-filling [63]

Flood-filling (figure 2.7) is another region-based segmentation algorithm that groups regions based on the similarity between each connected pixel in that group. The flood-filling algorithm in 3D Slicer requires the user to click within the region they

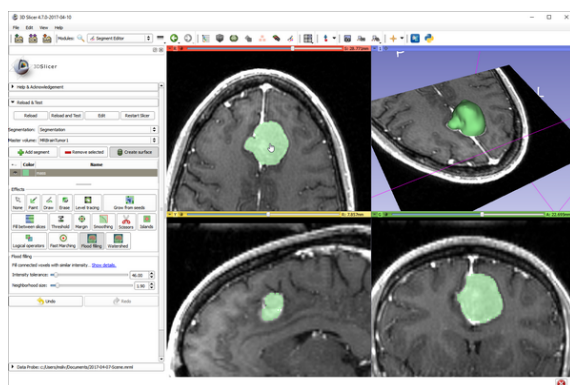
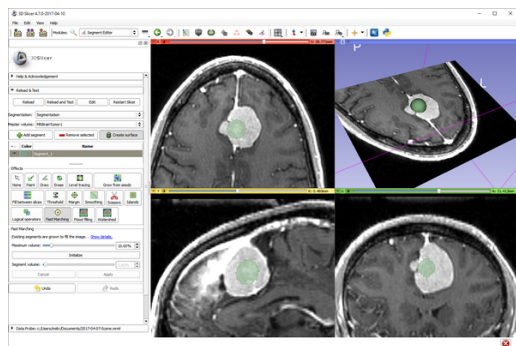
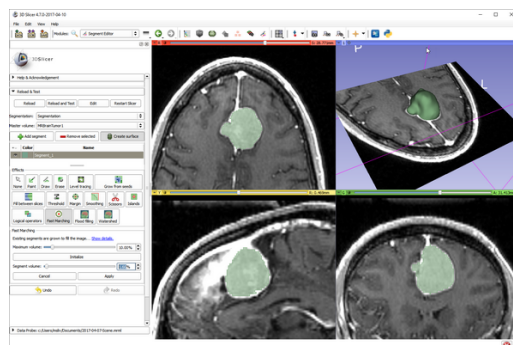


Figure 2.7: The result of flood-filling by clicking on the tumor. Taken from [62].

desire to segment, and the algorithm automatically segments the region based on the most similarity amongst connected pixels. The threshold for the sensitivity of these pixels can be changed to increase or decrease the number of pixels to include.



(a) The input image and user annotations to apply fast-marching segmentation



(b) The result of fast-marching segmentation

Figure 2.8: Input and output for fast-marching segmentation using 3D Slicer, taken from [62].

### 2.3.3 Fast-marching [64], [65]

The fast-marching segmentation technique is based off of its numerical namesake which determines a region as it grows from a start (seed) point. It leverages the concept of an optimal control solution to a bounded numerical method, by constantly

updating the distance of the current point from the seed point for each point in the image. 3D Slicer allows the user to place a seed-point (figure 2.8a) to begin segmentation and automatically updates the visual display with the segmented region (figure 2.8b).

### 2.3.4 Deep Learning in Medical Imaging

Starting in the 1960s, algorithms to interpret radiographic images were developed followed by Computer Based Diagnostic (CBD) [66] methods in the 1980s. CBD algorithms were mainly used for cancer detection, CT, and ultrasound to name a few. With the advent of Convolutional Neural Networks (CNNs), Shift-Invariant Artificial Neural Networks (ANNs), and Massively Trained Artificial Neural Networks (MTANNs) which were all inspired by neocognitron architectures [67] and are similar to modern day deep networks, researchers began exploring the applicability of DL based techniques for Medical Image analysis [68].

With the growing sources of medical imaging data collected and shared by various healthcare providers, modern day AI based tools such as Deep Networks are being designed at a staggering rate to improve the diagnosis of various diseases [69].

The applications of interest to this thesis are specifically segmentation of medical images using CNNs. There have been a variety of state of the art CNNs designed to segment different organs [70] like lungs[71], liver[72], skull[73], and the brain [74], [75].

### 2.3.5 NVIDIA Clara [76]

NVIDIA™ is at the forefront of designing both hardware and software to support the growing trend of AI. It's GPU based acceleration (CUDA)[77] enables the use of thousands of cores to increase the speed of matrix multiplications which



is paramount for training DL models faster. One of NVIDIA’s recent projects is known as Clara which attempts to democratize the use of AI for healthcare applications by providing Application Programming Interfaces (APIs) and Software Development Kits (SDKs) for imaging, genomics, smart sensors, etc.

### **2.3.5.1 NVIDIA Artificial Intelligence Assisted Annotation [78]**

The NVIDIA Artificial Intelligence Assisted Annotation (AIAA) tool is one of the many tools in the NVIDIA Clara suite mentioned before. It provides a server-client SDK with the server running a docker instance of the AIAA inference engine and the client providing endpoints to communicate with the server using the Hyper Text Transfer Protocol (HTTP). The surgical planning toolkit uses a pre-trained brain tumor annotation model which has been trained on T1 contrast weighted MR images and is readily available to use with the AIAA tool. The model is available under the category of DEXTR3D which stands for Deep Extreme cut in 3D, a 3D segmentation approach which combines the works presented by authors in [79], [80]. DEXTR3D requires a minimum of 6 manually placed markers which serve as the extremum of the region that the user desires to segment. The model creates a heatmap using Gaussians that are centered around each extreme point. The heatmap is concatenated and used as the input to the CNN. The final output segments the desired region.

## **2.4 NeuroRobot Overview**

The NeuroRobot is a 7 Degree of Freedom (DOF) robot designed completely of MRI-compatible materials with piezoelectric actuators and precise encoders, all-together tasked to perform surgical interventions in the brain such as Deep Brain



Figure 2.9: A Leksell stereotactic frame [10], commonly used to perform various neurosurgeries. This mechanism offers a Remote Center of Motion (RCM) to attain precise control while performing minute surgical tasks. The image shown is courtesy of Elekta Leksell Stereotactic System®(Elekta Instruments, Stockholm, Sweden)

Stimulation (DBS) and Tumor Ablation (Neuroablation). A typical procedure with the NeuroRobot can be broken down into 3 steps, **1)** Registration of the robot - the process of transforming the imaging coordinate frame to the robot coordinate frame using Acoustic MedSystems™Theravision [81], [82] on T1 FLAIR sagittal scans of the *Z-Frame* [83] which makes the base of the robot (figure 2.11) and obtaining the homogeneous transformation matrix for this operation, **2)** Targeting - The process of identifying the EP and TP and navigating the robot to reach the TP via the EP, and **3)** Intervention - The desired surgical intervention which could either be Neuroablation of a tumor or Deep Brain Stimulation.

### 2.4.1 Robot Design and Kinematics

Each conformal tumor ablation procedure using the NeuroRobot follows a systemic pre-operative surgical planning approach to ensure safe and accurate operation to maximize the surgical outcome. This requires an accurate knowledge of the robot's kinematics.

The robot kinematics are similar to that of a Leksell stereotactic frame (figure 2.9) and consist of 7 DOFs with a Remote Center of Motion (RCM) (figure 2.10, and 2.11) where three prismatic axis at the base determine the positioning of the RCM point and 2 rotational axis are used for orienting the probe into a desired pose. Finally, the last two DoFs including a prismatic and a revolute joint are used for probe insertion and rotation (figure 2.11) respectively.

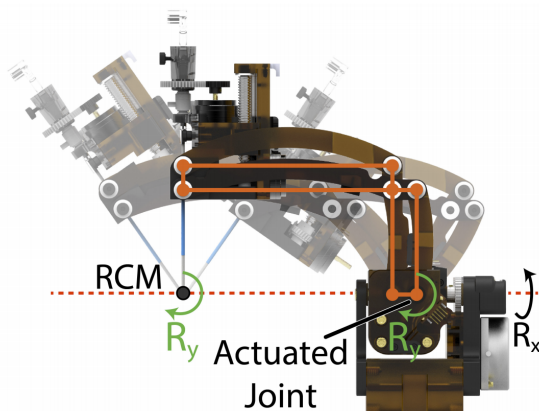


Figure 2.10: Side view of the NeuroRobot showing the Remote Center of Motion (RCM) along with the joints that are actuated to achieve this motion.[29]

### 2.4.2 Controlling the NeuroRobot [84]

#### 2.4.2.1 Hardware

As the robot functions within the bore of the MR scanner, the electronics for the robot need to be distanced from the high strength magnetic fields and proper mag-

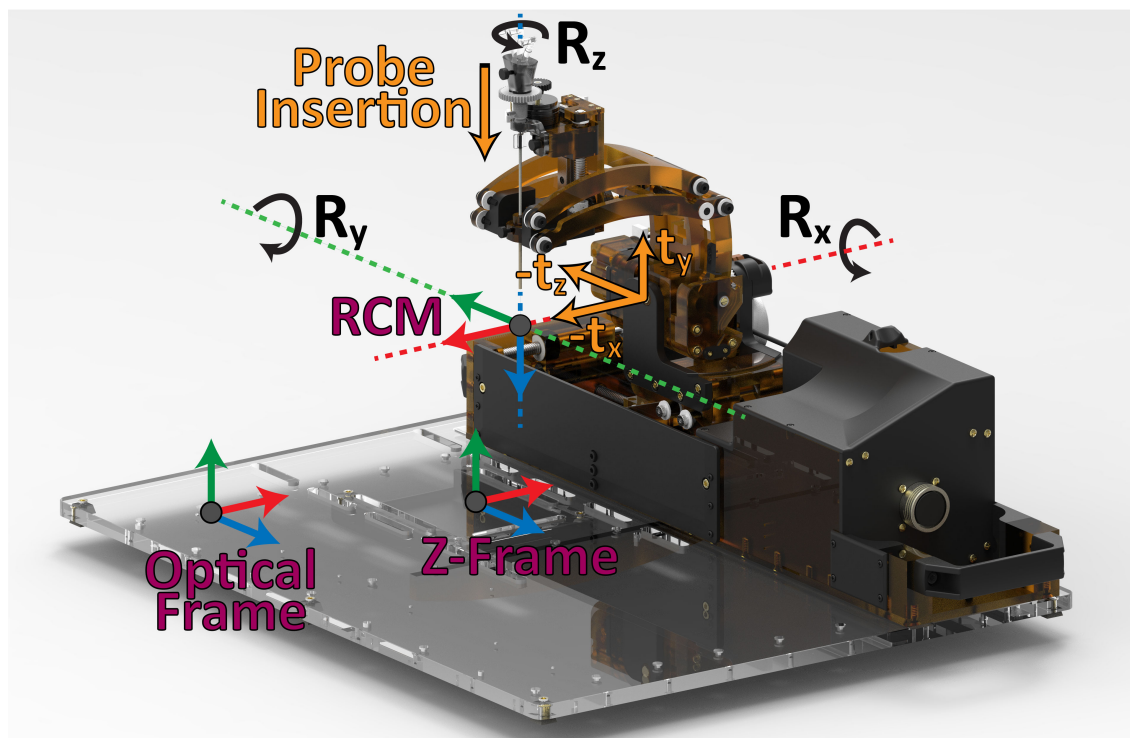


Figure 2.11: A labeled image of the NeuroRobot showing the coordinate frames, and axis of probe insertion along with the rotation DOFs to control the rotation of the probe. [29]



Figure 2.12: The robot control box which hosts all the electronic components of the NeuroRobot. The thick black cable coming out of the image is a shielded robot cable. The image was directly taken from [84].

netic shielding needs to take place to prevent interference. The control box (figure 2.12) contains the robot's power supplies, daughter cards, backplane, and breakout board and interface with the robot using a shielded robot cable.

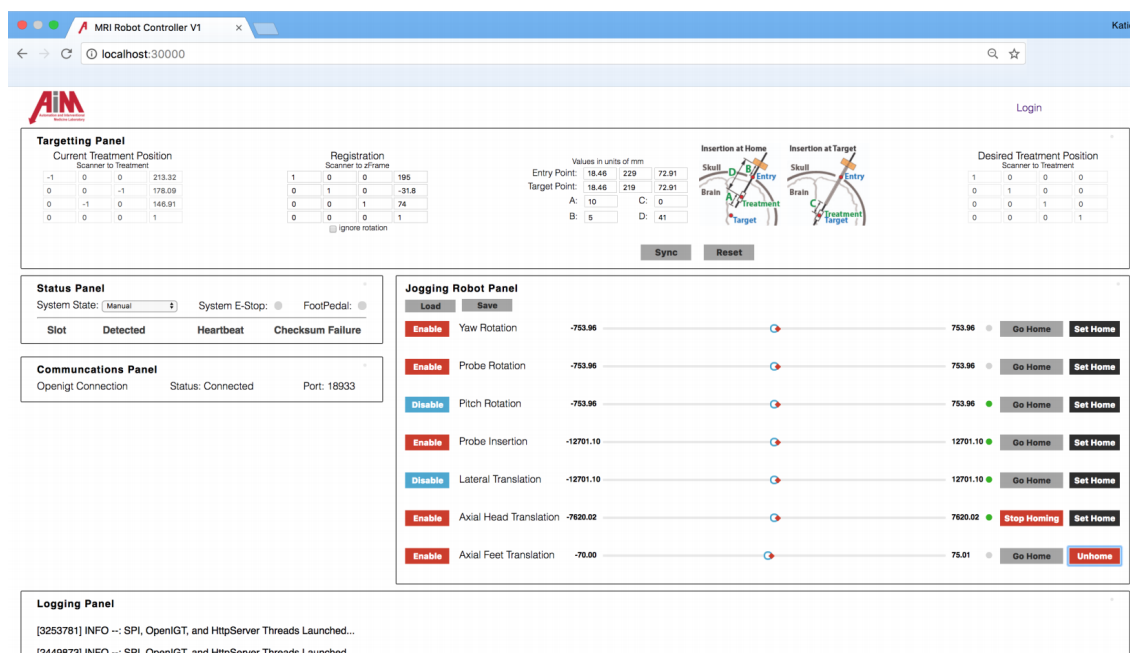


Figure 2.13: The robot control interface developed in ReactJS that communicates with the robot control box using HTTP requests. The image was directly taken from [84].

### 2.4.2.2 Software

The robot control software is primarily written in *C++* and is designed to provide the logic behind actuation and control. The code runs on a real-time Linux OS on an ARM Cortex A9 processor which is situated inside the robot’s control box. The software also provides endpoints to exchange data with the external world using OpenIGTLink [28], and serves a GUI (figure 2.13 which was developed in ReactJS and sends HTTP requests to modify configurations of the robot.

## 2.5 Neuroablation

The NeuroRobot’s structure is primarily plastic with a few components made of non-ferromagnetic materials to provided added support and actuation. The robot’s base hosts a *Z-frame* and is used for the registration process of the robot in the scanner’s

imaging frame. For Neuroablation experiments, a NBTU probe (figure 2.14) is used to perform the ablation of tumors[30], [84]. The probe and applicator was designed by Acoustic MedSystems™ and has been used previously as the applicator for animal studies [85], [86]. A PZT-4 transducer is used as the ultrasonic heating element and can produce 60, 90, 180, and 270 heating patterns.

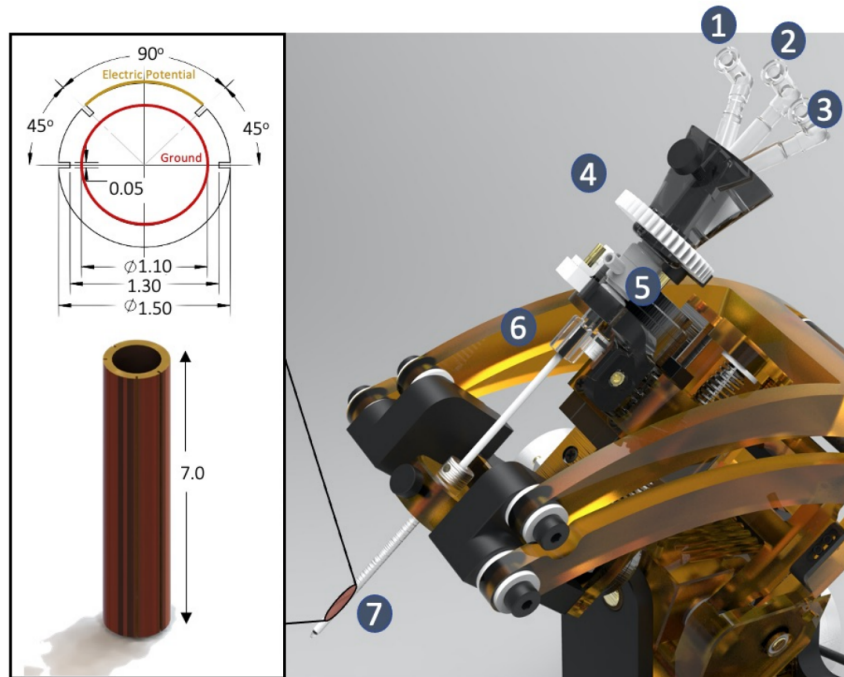


Figure 2.14: The NBTU probe developed by Acoustic MedSystems™. 1) Ablation element connector, 2) Coolant inlet, 3) Tracking coil interface connector, 4) Rotation control gear, 5) Coolant outlet, 6) Probe fixture, 7) Ultrasound element with sleeve. (Left) The PZT-4 transducer schematic which is used within the applicator. The image and labels were taken from [84]

The procedure undertaken for neuroablation requires preoperative images of the patient to identify the specifics of the tumor and a treatment area is identified. A cranial burr-hole is drilled to allow access to the treatment area after which the robot and patient are placed in MR scanner's bore and the robot is navigated to the EP location. The probe is then inserted into the brain and the NBTU probe is manipulated in such a way as to ablate the desired treatment area while being

tracked in the MR scanner during the procedure.

## 2.6 Burr-hole Detection using Medical Imaging

The problem of burr-hole detection *in-situ* MR imaging is both unique and challenging. During a normal procedure, once the burr-hole is drilled the surgeon covers the hole before placing the subject on the scanner's bed. This tends to make the burr-hole look like a closed cavity near the surface of the skull in the MRI images (figure 2.15). This section covers a few techniques that might improve the image quality for burr-hole detection, with some techniques already attempted by the researchers working on the NeuroRobot project.

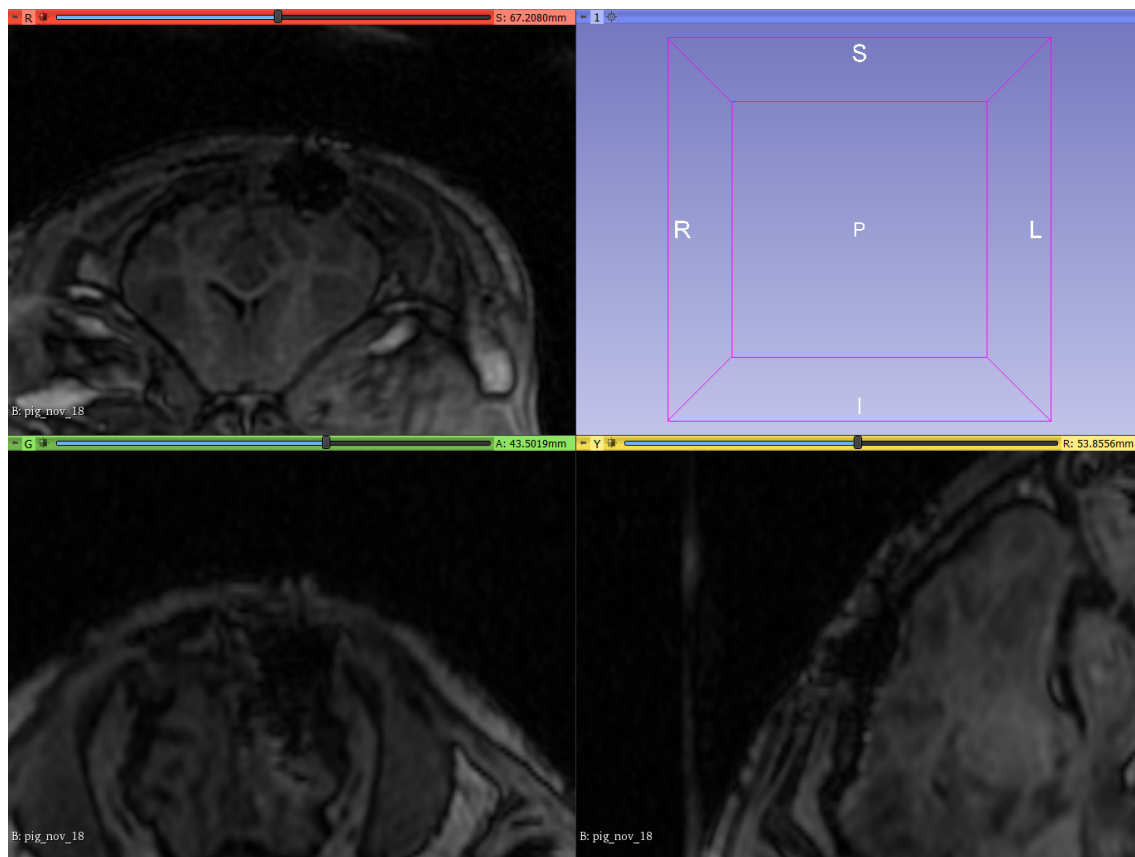


Figure 2.15: Burr-hole as seen in an MRI image of a pig.

- **Gadolinium Contrast** [87], [88], [89]: Researchers have previously used Gadolinium (Gd) as a contrast agent to attempt to improve the quality of the MRI image. Gd was inserted in the burr-hole in the form of a gelatin insert before the imaging sequences. Gd has seven unpaired electrons, thus enhancing the quality of the image. Unfortunately, complications like the gel melting caused an edema with the application of this contrast agent and further attempts to use this technique were halted.
- **MR-compatible markers** [90]: These markers are visible as distinct objects in an MR image and have an adhesive to help attach them to the surface of the skin during a scanning sequence. Researchers in [90] drill screws into the skull to try and validate stereotactic systems. Researchers on the NeuroRobot project, attempted to paste several markers around the region of the burr-hole, however, the resulting images remained hard to help identify the exact center and volume of the burr-holes.
- **Black Bone Imaging** [89]: Black bone imaging is a novel sequence and is an alternative to CT scans for bone imaging. These sequences use high-resolution fast echo sequences to minimize contrast with soft tissue. It is common to see these sequences used in cochlear implant surgeries, craniofacial disorders, etc. This method is yet to be tested for the NeuroRobot's use case to detect burr-holes.

To overcome this challenge in burr-hole detection, this project attempts to segment the black air cavity (figure 2.15) and identify the segmented region as the effective burr-hole.



# Chapter 3

## Surgical Planning Toolkit

This chapter presents the application and system design requirements, describes the architecture of the system and its components and provides an in-depth overview of the various algorithms and tools used.

### 3.1 Requirements

The primary function for this software is to *“Assist the operating surgeon by augmenting their field of view and in selecting the optimal insertion and treatment locations while prioritizing safety and speed”*.

#### 3.1.1 Application Requirements

The following is a detailed list of requirements to achieve the aforementioned vision.

- Augment the robot’s reachable workspace on the MR image.
- Identify the burr-hole and the optimal EP.
- Generate the treatment sub-workspace given an EP.

- Check whether the user-placed TP is reachable.
- If the EP is automatically identified, allow the surgeon to move the EP around if its location is unsatisfactory.

### 3.1.2 Design Requirements

As this Surgical Planning Toolkit is designed as an alternative tool to Bove’s research [33], it is necessary to meet the same design requirements posed by Bove. These requirements have been listed below for clarity.

- The toolkit should be cross-platform compatible.
- The toolkit should be portable.
- The toolkit should be user friendly.
- The toolkit should be developer friendly.

Additional design requirements that were not met by Bove’s research and have been considered are listed below.

- The tools offered by this toolkit should be validated for safety, and the accuracy of the results should be measured and within acceptable bounds.
- The toolkit should offer real-time planning for it to be used in a clinical trial.

## 3.2 Architecture

Figure 3.1 shows the layout of the architecture for the surgical planning toolkit. The interactive GUI allows the user to create instances of MRML nodes that are consumed by this module, such as a *vtkMRMLSegmentationNode* which stores

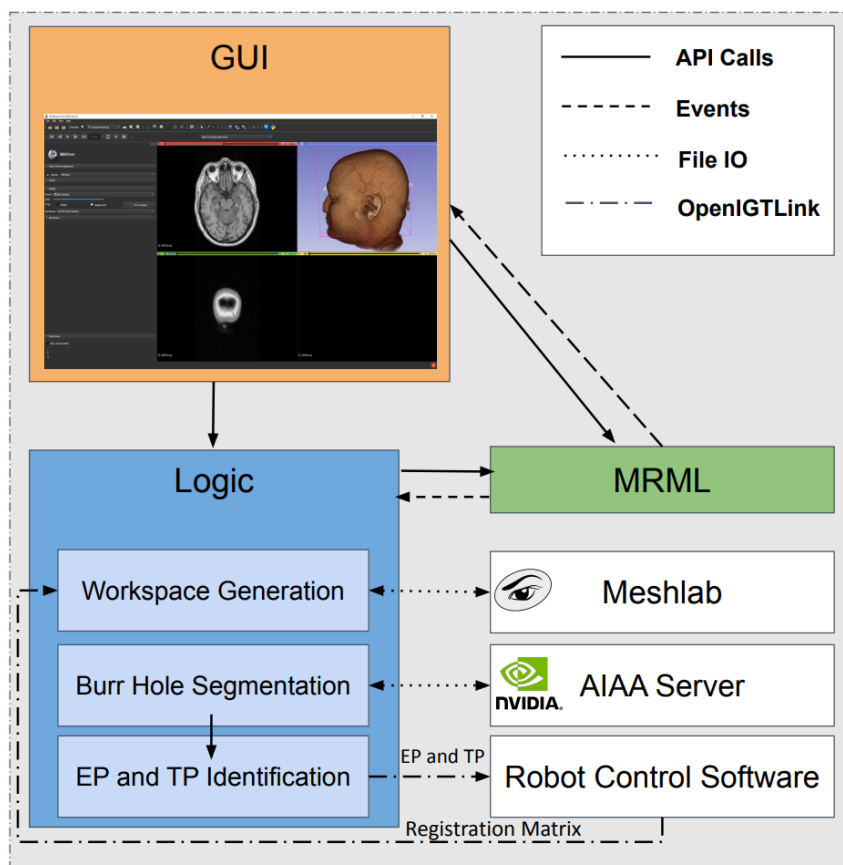


Figure 3.1: The Robot Control Software pushes the registration matrix to the Logic using OpenIGTLink after the registration sequence is complete<sup>1</sup>. The GUI communicates with the Logic and Medical Reality Modelling Language (MRML) node using API calls, The Logic in turn communicates with the MRML Node using API calls and external modules like the Nvidia AIAA Server and Meshlab Server through File IO. The MRML Node can communicate with both the GUI and Logic by triggering events. Once the EP and TP have been identified, the coordinates can be pushed to the robot control software using OpenIGTLink.

properties of a 3D segmentation, as well as parameters required for surgical planning in the *vtkMRMLSurgicalPlanningNode*. These parameters include the probe specifications, registration matrix, burr-hole boundary markups, burr-hole center, EP, TP, and other variables that retain the state of the GUI. Interacting with the GUI also signals a logic call that performs the desired operation and updated the MRML node accordingly. The logic communicates with the Nvidia AIAA Server

through a `C++` client which in turn serves a compressed MRI image file over an HTTP push. The logic also communicates with the Meshlab Server by passing the workspace as a point set XYZ file to the server and loading the generated mesh as a polydata (`.ply`) file. Once the EP and TP have been selected, the Logic pushes the coordinates to the Robot Control Software to target the tumor and begin the procedure for ablation. Additionally, the Robot Control Software can also push the registration matrix to the Logic using OpenIGTLink at the start of the planning procedure.

### 3.2.1 Graphical User Interface

The GUI enables the surgeon to interact with the surgical planning toolkit. The GUI is structured like a stateflow with different components of the toolkit clustered into sections based on their utility. Utilities that depend on other components are only made available if the earlier steps have been completed, such as detecting the burr-hole (figure 3.2(3.3)) depends on the application to be connected to the AIAA server (figure 3.2(3.1)) and the push button is only made available once the application has connected to the AIAA server. By default, the `SurgicalPlanningMRMLNode` is created upon loading the Slicer module (Figure 3.2(0)). Once the DICOM image has been received from the MRI scanner computer, the volume can be loaded into the module using the Input Volume selector (figure 3.2(1.1)). The volume is automatically rendered upon selection using GPU Ray casting. Rendering the volume uses Slicer's internal `VolumeRendering` module and automatically populates an `AnnotationROI` node for the rendered volume (figure 3.2(1.2)). After loading the input volume, the remaining fields become enabled and the surgeon can continue with the procedure. The input fields in figure 3.2(2.1) have default probe specifications for the surgeon's convenience but can be modified to probe specifications of the probe

0 Parameter Node SurgicalPlanningMRMLNode

**1. Input Settings**

1.1 Input Volume pig\_nov\_18

1.2 ROI ROI\_pig\_nov\_18 Visibility

▶ Volume Rendering Preset Settings

**2. Robot Workspace Settings**

▶ Probe Specifications

▶ Probe Specifications Help

2.1

A 7.00
B 5.00
C 0.00
D 33.00

▶ Registration Matrix

1.000	0.000	0.000	3.590
0.000	1.000	0.000	-131.750
0.000	0.000	1.000	-15.380
0.000	0.000	0.000	1.000

2.2

2.3 Workspace Mesh GeneralWorkspace Visibility

2.4 Entry Point WS EntryPointWorkspace Visibility

2.5

Generate Workspace
Generate Entry Point WS

**3. Burr Hole Detection**

▶ Burr Hole Configuration

3.1 AIAA Server http://127.0.0.1:8123/ Success

100%

3.2 Boundary Markups BHPoints

↑

⋮

3.3 Detect BurrHole

3.4 Burr Hole Segmentation BurrHoleSegmentation Visibility

**4. Entry Point and Target Point Selection**

4.1 Entry Point (EP) EntryPoint

↑

⋮

4.2 Sub Workspace Segmentation SubWorkspace Visibility

4.3 Generate Sub Workspace

4.4 Target Point (TP) TargetPoint

↑

⋮

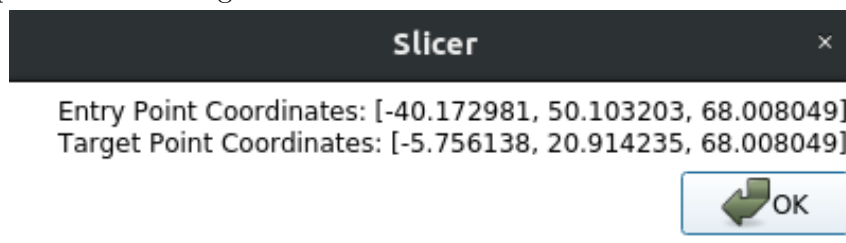
Figure 3.2: Labelled image of the GUI for the surgical planning toolkit. 0) SurgicalPlanningMRMLNode selector. 1.1) Selector for input MR volume node. 1.2) Selector for Annotation ROI node of rendered input volume. 2.1) Input fields for Probe specifications. 2.2) Input field for Registration Matrix. 2.3) Selector for general workspace segmentation node. 2.4) Selector for entry point segmentation node. 2.5) Push buttons to generate workspace meshes. 3.1) Input field to connect to AIAA server. 3.2) Selector and Markup node for burr-hole boundary markups node. 3.3) Push button to detect burr-hole. 3.4) Selector for burr-hole segmentation node. 4.1) Selector and Markup node for EP markups node. 4.2) Selector for sub-workspace segmentation node. 4.3) Push button to generate sub-workspace mesh. 4.4) Selector and Markup node for TP markups node.

that's in use. The surgeon then enters the Registration Matrix from the initial registration scan to align the workspaces and navigation markups (figure 3.2(2.2)). Figure 3.2(2.3, 2.4) show selectors for the workspace segmentation nodes along with respective push buttons that generate the workspaces (figure 3.2(2.5)). To use the assisted EP placement offered by this toolkit, the surgeon needs to connect to the AIAA server which defaults to the localhost (figure 3.2(3.1)). Once connected successfully, the next few interactive fields are enabled. First, the surgeon selects a markups node and places fiducial markers along the boundary of the burr-hole (figure 3.2(3.2)) and then allows the AIAA tool to automatically segment the region of the burr-hole (figure 3.2(3.3, 3.4)) and detect its center which is labeled as the EP (figure 3.2(4.1)). This fiducial marker can be manually moved around if the resulting location is unsatisfactory to the surgeon, or the entire process of segmenting and detecting the burr-hole and EP respectively can be repeated by simply pressing the *Detect burr-hole* button (figure 3.2(3.3)) again after moving the fiducial markers around. If the EP lies beyond the Total Reachable Workspace (TRW) of the robot, a popup (figure 3.3a) is raised warning the surgeon that the EP is not reachable and either the EP should be moved, or subject should be moved on the MRI scanner bed to increase the overlap between the total reachable workspace and the subject's

head. Next, the surgeon can generate the sub-workspace (figure 3.2(4.3)) after selecting a sub-workspace segmentation node (figure 3.2(4.2)). Finally, the surgeon can select a TP markup node (figure 3.2(4.4)) and place the TP within the generated sub-workspace. Finally, a popup (figure 3.3b) is raised to display the final EP and TP coordinates in RAS coordinates, with respect to the NeuroRobot's reference frame.



(a) Popup to warn the surgeon that the EP lies outside the total reachable workspace.



(b) Popup to indicate the selected EP and TP coordinates in RAS space with respect to the robots frame.

Figure 3.3: Popups created by the GUI and Logic to direct the surgeon's attention to an important step.

### 3.2.2 Medical Reality Modeling Language Node

The MRML Node acts as the *Model* in the MVC like design pattern that Slicer uses. It stores state of the application and can be saved to a file, or loaded from a file when required. The representation of the MRML node is XML and the Slicer API provides convenient helper functions to cast primitive types to XML like the `vtkMRMLWriteXMLBooleanMacro(xmlAttributeName, propertyName)` for booleans, `vtkMRMLWriteXMLFloatMacro(xmlAttributeName, propertyName)` for floats,

etc.. Listing 3.1 shows the class definition for a `struct` that stores information of the burr-hole after the toolkit has segmented and identified the center. The setters defined in this class provide public functions to set the center of the burr-hole in RAS coordinates in the imaging frame, set the radius of the burr-hole, and set the model of the drill bit that was used to drill the burr-hole as a `vtkMRMLModelNode*`.

Listing 3.1: Burr-hole parameters class specification

```
class BurrHoleParameters
{
public :
    // Setters
    /**
     * @brief Set the Center object
     *
     * @param vtkVector3d center
     */
    void setCenter(vtkVector3d center);
    /**
     * @brief Set the Radius object
     *
     * @param double radius
     */
    void setRadius(double radius);
    /**
     * @brief Set the Drill Bit object
     *
     * @param vtkMRMLModelNode* drill_bit
     */
    void setDrillBit(vtkMRMLModelNode* drill_bit);

    // Getters
    /**
     * @brief Get the Center object
     *
     * @return vtkVector3d center
     */
    vtkVector3d getCenter() const;
    /**
     * @brief Get the Radius object
     *
     * @return double radius
     */
    double getRadius() const;
    /**
     * @brief Get the Drill Bit object
     *
     * @return vtkMRMLModelNode* drill_bit
     */

```



```
    */
    vtkMRMLModelNode* getDrillBit () const ;

private :
    vtkVector3d      _center ;
    double          _radius ;
    vtkMRMLModelNode* _drill_bit ;
};
```

In addition to storing the state of the module, the MRML node also handles events which can be triggered by the Logic or the GUI. The GUI and Logic can add *observers* which serve as callbacks/handlers for the events defined in the MRML Node. Mostly, these events are triggered on modification of consumed nodes, such as a modification of the workspace segmentation nodes triggers the `vtkMRMLSurgicalPlanningNode` to update its state. Listing 3.2 shows a custom event that triggers when any Markup that is being observed by the `vtkMRMLSurgicalPlanningNode` is modified (moved, renamed, created, deleted, etc). The listing also shows how observers are defined and events are handled.

Listing 3.2: `vtkMRMLSurgicalPlanningNode` class specification. This listing focuses on discussing the event handling by the MRML Node.

```
/**
 * @brief MRML Node for the Surgical Planning Toolkit,
 * inherits vtkMRMLNode
 *
 */
class VTK_SLICER_SurgicalPlanning_MODULE_MRML_EXPORT
    vtkMRMLSurgicalPlanningNode : public vtkMRMLNode
{
.
.
.
public :
    enum Events
    {
        /// MarkupsPositionModifiedEvent is called when markup
        /// point positions are modified. This make it easier
        /// for logic or other classes to observe any changes
        /// in input data. vtkCommand::UserEvent + 777 is just
        /// a random value that is very unlikely to be used for
        /// anything else in this class
        MarkupsPositionModifiedEvent =
            vtkCommand::UserEvent + 777
    }
};
```

```
};

/**
 * @brief Set the And Observe InputVolumeNodeID
 *
 * @param const char* inputNodeId
 */
void SetAndObserveInputVolumeNodeID(
    const char* inputNodeId);

/**
 * @brief Set the And Observe AnnotationROINodeID
 *
 * @param const char* annotationROINodeId
 */
void SetAndObserveAnnotationROINodeID(
    const char* annotationROINodeId);

/**
 * @brief Set and observe the
 *      WorkspaceMeshSegmentationNodeID
 *
 * @param const char* workspaceMeshSegmentationNodeId
 */
void SetAndObserveWorkspaceMeshSegmentationNodeID(
    const char* workspaceMeshSegmentationNodeId);

/**
 * @brief Set and observe the
 *      EPWorkspaceMeshSegmentationNodeID
 *
 * @param const char* ePWorkspaceMeshSegmentationNodeId
 */
void SetAndObserveEPWorkspaceMeshSegmentationNodeID(
    const char* ePWorkspaceMeshSegmentationNodeId);

/**
 * @brief Set and observe the
 *      SubWorkspaceMeshSegmentationNodeID
 *
 * @param const char* SubWorkspaceMeshSegmentationNodeId
 */
void SetAndObserveSubWorkspaceMeshSegmentationNodeID(
    const char* SubWorkspaceMeshSegmentationNodeId);

/**
 * @brief Set the And Observe BurrHoleSegmentationNodeID
 *
 * @param const char* burrHoleSegmentationNodeId
 */
void SetAndObserveBurrHoleSegmentationNodeID(
    const char* burrHoleSegmentationNodeId);

/**
 * @brief Set the And ObserveBHExtremePointNodeID
 *
 * @param const char* bHExtremePointNodeId
 */
void SetAndObserveBHExtremePointNodeID(
```

```
    const char* bHExtremePointNodeId);  
/**  
 * @brief Set the And Observe EntryPointNodeID  
 *  
 * @param const char* entryPointNodeId  
 */  
void SetAndObserveEntryPointNodeID(  
    const char* entryPointNodeId);  
/**  
 * @brief Set the And Observe TargetPointNodeID  
 *  
 * @param const char* targetPointNodeId  
 */  
void SetAndObserveTargetPointNodeID(  
    const char* targetPointNodeId);  
/**  
 * @brief Process any event that is triggered, pass the  
 * caller (GUI/Logic), event id, and data to be handled  
 * by the end.  
 *  
 * @param vtkObject* caller  
 * @param unsigned long event  
 * @param void* callData  
 */  
void ProcessMRMLEvents(vtkObject* caller, unsigned  
    long event, void* callData) VTK_OVERRIDE;  
.  
.  
};
```

### 3.2.3 Logic

The Module's logic acts as the Controller in the MVC like architecture that Slicer follows. It serves the purpose of implementing *algorithms* to process the volume and model nodes that have been consumed by the `vtkMRMLSurgicalPlanningNode`. The logic communicates with both the GUI and MRML Node using API calls and events. The key algorithms that are implemented in the logic are to generate the workspaces and identify the burr-hole.

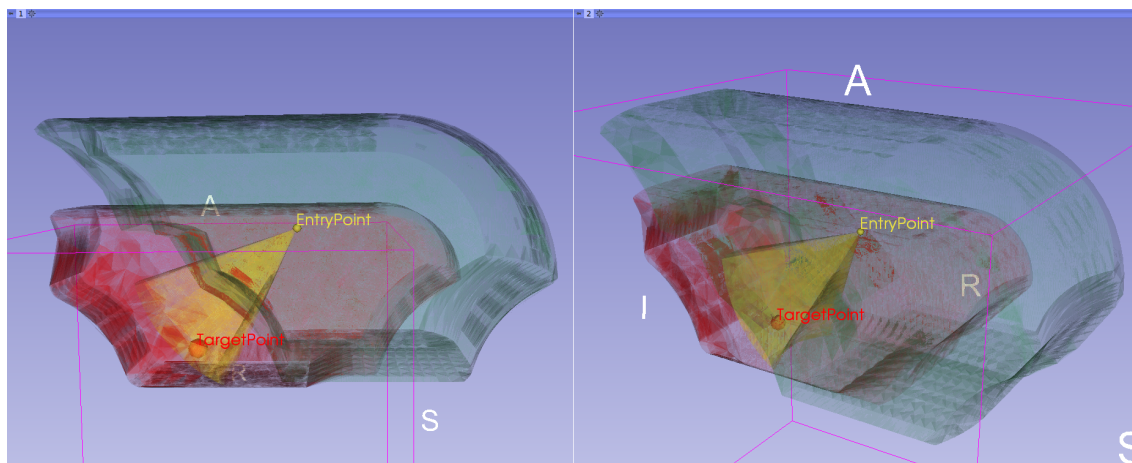


Figure 3.4: NeuroRobot’s workspaces generated by the Surgical Planning toolkit. The green workspace is the EP workspace, representing all possible locations where an EP can be placed. The red workspace is represents the total reachable workspace, which includes all the possible locations where the instrument’s treatment tip can reach. Finally, the yellow workspace is the sub (or treatment) workspace, representing all the locations where the instrument’s treatment tip can reach given the probe enters from an EP.

### 3.2.3.1 Workspace Generation

Complex kinematics of the NeuroRobot and having an RCM mechanism create an irregular workspace shape, which make it difficult to estimate its shape and position without visual assistance. Additionally, the robot’s workspace is dependent on the physical dimensions of the treatment instrument (e.g. an interstitial ablation probe, needle, DBS lead) and the safety margins set by the surgeon that define the distance of the robot from the EP. In order to assist the surgeon with the selection of valid EP and TP, this toolkit incorporates methods for generating different workspaces namely entry, total reachable, and sub workspace (figure 3.4). Each of these workspaces provide an essential piece of information to the surgeon regarding the robot’s reach relative to the desired intervention target within the brain. Each workspace is visualized and augmented as both a 3D volumetric object and a 2D geometric representation on the slices of the MRI image and the surgeon has the

ability to toggle the visualization of each of these workspaces on or off. The following is the description of each of these workspaces and their role in the surgical planning task of the NeuroRobot.

- *Entry Point Workspace (EPW)*: This workspace represents all possible locations where an EP can be placed. Thus, in cases where the burr-hole falls outside of this workspace, the surgeon can re-position the patient to a desired location to ensure proper coverage of the burr-hole by this workspace.
- *Total Reachable Workspace (TRW)*: The idea behind generating the TRW is to visualize all possible points where the robot can effectively deliver treatment. The treatment region can be configured in the module for a particular instrument, such as the tip of a biopsy needle or the electrode of a DBS lead as the clinical application at hand dictates. The surgeon can use this workspace to determine if the desired target region is reachable by the treatment tip and take corrective actions if needed.
- *Sub Workspace (SW)*: This workspace is generated once an EP is placed on the MRI image. The volume that this workspace represents includes all possible location that the treatment can reach given a desired EP. Therefore, this workspace assists the surgeon with selecting the TP that is kinematically valid for the robot to reach. As mentioned earlier, the initial EP is generated based on the calculations of centroid of the burr-hole segmentation. So, in order to generate the SW, the surgeon can either use the recommended EP or relocate the EP and generate the corresponding SW. The visual overlay of this workspace could be an intuitive and straightforward method for surgeons to asses the range of reachable targets, and consequently, increase the accuracy

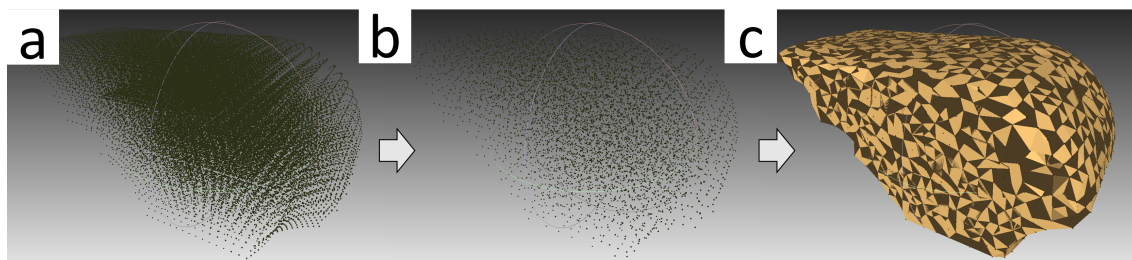


Figure 3.5: Mesh generation using the MeshLab server. a) Point set is passed to the MeshLab Server, b) Filters are applied to the point set, c) 3D mesh is generated and loaded inside the module.

and decrease the time of the procedure.

The generation of the robot’s workspace can be achieved by employing a combination of the robot’s Forward Kinematics (FK) and Inverse Kinematics (IK) models. By iterating through joint values of the robot and passing them to the FK model, an array of points is obtained that corresponds to the location of the robot’s end-effector with respect to the robot’s fixed base frame (Z-frame). The TRW and EPW are generated using only the robot’s FK, while the SW is generated using a combination of the robot’s FK and IK. The algorithm for the SW, first checks a subset of the points in space that are within a specific distance from the selected EP and stores them in an array. In the next step, the stored points are checked using the robot’s IK and the points which result in valid configurations are saved. In the final step, the stored validated array is given to the robot’s FK and a point set corresponding to the reachable points is created.

### 3.2.3.2 Workspace Visualization

To visualize the generated workspaces in the Slicer, each point set needs to be converted into a 3D mesh model. This can be done using surface generation algorithms that have been implemented by various visualization toolkits. The Surgical Planning Toolkit investigated and profiled two approaches to visualize the mesh, **1)** Delauney

triangulation [91] with alpha shapes [92] based concave mesh generation, and **2**) Powercrust Surface Reconstruction [93]. The approach using alpha shapes yielded a better workspace and was more efficient and was implemented in the final version of the toolkit.

This toolkit uses MeshLab[58] to perform the rendering of the workspace. Meshlab has a feature that enables batch processing using MeshLab filters without using its GUI and through a system executable called the MeshLab server. MeshLab server requires three inputs 1) an XML script which contains all the steps for applying filters to the desired mesh (listing 3.3), 2) an input file, 3) an output file. The XML script can be conveniently generated by applying all the desired filters on a sample mesh and saving all the steps into an XML file using Meshlab. The input file to the MeshLab server is a file containing the generated point set in the XYZ format and the output is the 3D mesh model in the PLY format. Each time a request for a workspace is generated in the module, a system call will evoke the MeshLab server and passes the corresponding point set to it and returns the generated 3D mesh model to the Slicer and visualizes it.

Listing 3.3: XML script which contains all the steps for applying filters to the desired mesh

```
<!DOCTYPE FilterScript >
<FilterScript >
  <filter name="Point Cloud Simplification">
    <Param type="RichInt" value="1000"
      description="Number of samples"
      tooltip="The desired number of samples.
        The ray of the disk is calculated
        according to the sampling density."
      name="SampleNum"
    />
    <Param type="RichAbsPerc" value="0"
      description="Explicit Radius"
```

```
    min="0"
    tooltip="If not zero this parameter
    override the previous parameter to
    allow exact radius specification"
    max="265.615"
    name="Radius"
  />
  <Param type="RichBool" value="true"
    description="Best Sample Heuristic"
    tooltip="If true it will use a simple
    heuristic for choosing the samples. At
    a small cost (it can slow a bit the
    process) it usually improve the
    maximality of the generated sampling."
    name="BestSampleFlag"
  />
  <Param type="RichInt" value="10"
    description="Best Sample Pool Size"
    tooltip="Used only if the Best Sample
    Flag is true. It control the number of
    attempt that it makes to get the best
    sample. It is reasonable that it is
    smaller than the Montecarlo
    oversampling factor."
    name="BestSamplePool"
  />
  <Param type="RichBool" value="false"
    description="Exact number of samples"
    tooltip="If requested it will try to do
    a dicotomic search for the best poisson
    disk radius that will generate the
    requested number of samples with a
    tolerance of the 0.5%. Obviously it
    takes much longer."
    name="ExactNumFlag"
  />
</filter >
<filter name="Delete Current Mesh" />
```



```
<filter name="Alpha Complex/Shape">
  <Param type="RichAbsPerc" value="13.2808"
    description="Alpha value"
    min="0"
    tooltip="Compute the alpha value as
    percentage of the diagonal of the bbox"
    max="265.615"
    name="alpha"
  />
  <Param type="RichEnum" value="1"
    enum_val1="Alpha Shape"
    description="Get:"
    tooltip="Select the output. The Alpha
    Shape is the boundary of the Alpha Complex"
    enum_val0="Alpha Complex"
    enum_cardinality="2"
    name="Filtering"
  />
</filter >
</FilterScript >
```

Two primary filters were used to generate the mesh models. The first filter called *Point Cloud Simplification*, reduced the complexity of the point set while preserving its details by reducing the number of points in it. The second filter generated the 3D mesh from the simplified point set using the *Alpha Shapes* filter, which produce a concave 3D mesh representation of the point set. Figure 3.5 shows the rendered mesh.

### 3.2.3.3 Entry Point Identification

The task of detecting a burr hole is quite unique to our experiment. Traditional stereotactic surgery involves preoperative imaging for surgical planning after which a burr hole is drilled and the surgical intervention is performed with the help of a Leksell frame. In the case of surgical procedures performed using the NeuroRobot,

planning/re-planning is performed after the burr hole is drilled, thus, the location of the EP for the surgical instrument needs to be identified using just the MR image.

Listing 3.4: Watershed segmentation implemented in ITK.

```
// Create the filter object
using WatershedFilterType =
    itk::WatershedImageFilter<FloatImageType>;
WatershedFilterType::Pointer watershed =
    WatershedFilterType::New();

// Set the threshold and level
watershed->SetThreshold(threshold);
watershed->SetLevel(level);

// Pass the input volume after performing a
// gradientMagnitudeImageFilter on it
watershed->SetInput(
    gradientMagnitudeImageFilter->GetOutput()
);
// Update the algorithm to obtain the result
watershed->Update();
```

Listing 3.5: Flood-filling segmentation implemented in VTK and Slicer.

```
// The flood-filling filter is represented as a
// ConnectedThresholdFilter in the ITK Library.
// First the image is processed using a curvature flow
// image filter
using CurvatureFlowImageFilterType =
    itk::CurvatureFlowImageFilter<
        InternalImageType, InternalImageType
    >;
CurvatureFlowImageFilterType::Pointer smoothing =
    CurvatureFlowImageFilterType::New();
// Create an object of the flood-filling filter
using ConnectedFilterType =
    itk::ConnectedThresholdImageFilter<
        InternalImageType, InternalImageType
```

```
>;
ConnectedFilterType::Pointer connectedThreshold =
    ConnectedFilterType::New();

// Chain the ports
smoothing->SetInput(/*Image port*/);
connectedThreshold->SetInput(smoothing->GetOutput());

// Set the lower and upper threshold values to
// include pixels in between in the segment
connectedThreshold->SetLower(lowerThreshold);
connectedThreshold->SetUpper(upperThreshold);

// Update the filter to process the output
connectedThreshold->Update();
```

Listing 3.6: Fast-marching segmentation implemented in ITK.

```
// Preprocess the image by applying a gradient and sigmoid
// filter
GradientFilterType::Pointer gradientMagnitude =
    GradientFilterType::New();
SigmoidFilterType::Pointer sigmoid =
    SigmoidFilterType::New();
// Create a smoothing curvature anisotropic diffusion filter
SmoothingFilterType::Pointer smoothing =
    SmoothingFilterType::New();

// Create an object of the fast marching filter
using FastMarchingFilterType =
    itk::FastMarchingImageFilter<
        InternalImageType, InternalImageType
    >;
FastMarchingFilterType::Pointer fastMarching =
    FastMarchingFilterType::New();

// Chain all filters and process the output by running
// update
smoothing->SetInput(/*Image input*/);
```

```
gradientMagnitude->SetInput(smoothing->GetOutput());  
sigmoid->SetInput(gradientMagnitude->GetOutput());  
fastMarching->SetInput(sigmoid->GetOutput());  
  
fastMatching->Update();
```

This toolkit employs a semi-automated approach and assists the surgeon with segmenting the region where the burr-hole manifests as an air pocket by placing boundary points to demarcate the region. Before settling on using an AI based segmentation tool, various classical techniques like Watershed[61](listing 3.4), Flood-filling[63](listing 3.5), and Fast-marching[64](listing 3.6) were tested and compared with the results from the NVIDIA AIAA tool. Figure 3.6 shows a side-by-side rendering of the segments generated by these algorithms.

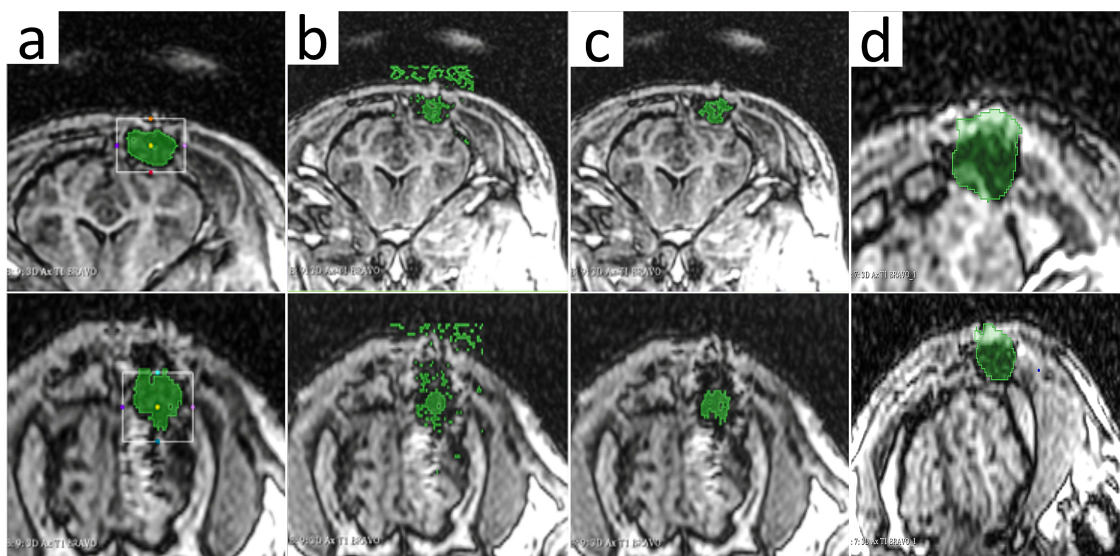


Figure 3.6: Qualitative comparison of the generated segments using (a) Flood-filling, (b) Watershed, (c) Fast-marching, and (d) NVIDIA AIAA Tool

After consulting with the operating surgeon, the toolkit integrated the NVIDIA AIAA Client and ran a local instance of the AIAA server on an NVIDIA 1060 6GB GPU. The AIAA server runs an instance of a modified version of Deep Extreme Cut [79] which uses extreme points on the borders of the object to accurately segment

the region within. The model used for the AIAA server is a pre-trained model used to identify brain tumors from T1c MR Images, annotate, and generate a 3D segmentation around them (*clara\_mri\_annotation\_brain\_tumors\_t1ce\_tc\_no\_amp* [94]). The pre-trained model was trained to segment the *tumor-core* in T1 weighted contrast images. The model includes a type of autoencoder regularization to annotate the segment and was trained on the BraTS dataset [95]. Additionally, the model was further trained on the MR Images of pig heads obtained from prior trials after a surgeon manually annotated each burr-hole in the image using the pre-trained model. This improved the segmentation results for burr-hole identification for later experiments.

The Surgical Planning Toolkit makes use of the NVIDIA AIAA client side in *C++* to communicate with the AIAA server. The API communicates with the server using HyperText Transfer Protocol (HTTP) requests. An overview of communicating with the server and identifying the burr-hole is as follows:

- 1) Initiate a session with the server for an image, to reduce the time taken during consecutive requests.
  - 1.1) Connect to the server, request list of available models from server.
  - 1.2) Transform from *RAS* coordinates to *IJK* coordinates, compress and save image as a temporary compressed *NiftI* (*.nii.gz*) file and store the filename.
  - 1.3) Create an HTTP PUT request and upload the compressed image to the server, receive a response form the server which contains the Session ID, store for future use.
- 2) Select the desired model.

- 3) Transform each extreme boundary point from *RAS* coordinates to *IJK* coordinates and add to a list of points.
- 4) Call the AIAA `dext3d` function and pass the model, extreme points, input file path, output file path, and sessionID.
  - 4.1) The AIAA Client creates a HTTP POST method to send the parameters to the server.
  - 4.2) The response from the server is used to identify any errors and debug exceptions.
- 5) Load the output segmented image as a Segmentation Node in Slicer, transform the node from *IJK* coordinates to *RAS* coordinates.
- 6) Remove small segments and keep the largest segments using island removal.
- 7) Identify the centroid of the segment using either a center of mass method or center of a bounding box method.

Listing 3.7: Island removal by searching for the largest segment using VTK.

```
// Create a connectivity filter object
vtkNew<vtkPolyDataConnectivityFilter> connectivityFilter;

// Pass the segmentation node
connectivityFilter->SetInputConnection(/*segmentation node*/);
// Set the extraction mode to find the largest region
connectivityFilter->SetExtractionModeToLargestRegion();
// set the segmentation node to the output of the filter
```

After segmentation of the burr hole, a filter is applied to remove all islands that may populate because of a noisy input signal and retain the main segmentation volume corresponding to the burr hole. This is done using the `vtkPolyDataConnectiv-`

`ityFilter` (listing 3.7) where the filter is set to search for the largest region in the segment. Next, the center of the burr-hole is identified by fitting a bounding box around the segmentation and using the center of the bounding box as the centroid of the burr hole volume. Since the segmented region is always convex, fitting a bounding box around the region is fast and provides an accurate centroid. The Cartesian coordinates of the centroid are then transformed from the image plane to the robot frame by multiplying it with the inverse of the robot's registration matrix and shown as the EP markup point on the MRI image.

# Chapter 4

## Experiments

Several experiments were carried out to test, validate, and improve the surgical planning toolkit over the course of the project, namely, **1)** Validation of generated workspaces through a phantom targeting trial, **2)** Validation of assisted Entry Point (EP) identification, **3)** User study to understand the benefits of assisted EP identification over manually identifying the EP, and finally **4)** A pig study to test the entire surgical planning workflow.

### 4.1 Validating generated workspaces

Generating accurate workspaces is critical to the procedure as the workspaces assist the surgeon in identifying if the desired entry and treatment locations lie within reach of the robot and ablation probe.

#### 4.1.1 Procedure

To test the workspace generation accuracy a targeting test was conducted on a gel-phantom with three embedded MRI-compatible markers (figure 4.1a). The goal of



the experiment was to assume a marker as the desired treatment location, generate the workspaces and attempt to reach the marker and validate the accuracy of the targeting performed. The process followed are listed below:

- 1) Prepare the gel-phantom and randomly place MRI-compatible markers during the forming process.
- 2) Register the robot in the imaging frame.
  - 2.1) Place the robot on the MRI scanner bed and perform a registration scan.
  - 2.2) Obtain the registration matrix for the robot by performing a Z-Frame registration[83].
- 3) Place the gel-phantom on the MR scanner bed.
- 4) Target a marker
  - 4.1) Generate the TRW and EPW and augment them on the 2D slices and 3D volume render of the gel-phantom.
  - 4.2) Check if the treatment location lies within the TRW.
  - 4.3) If the treatment lies outside the TRW, reposition the robot and repeat from step 2), else continue.
- 5) Identify and place the EP.
  - 5.1) Place an EP and generate the SW for that EP.
  - 5.2) If the treatment location doesn't lie within the SW, repeat step 4.1, else continue.
- 6) Place the TP on or closest to the desired treatment location.

- 7) Insert the probe and perform a validation scan to calculate the accuracy of the targeting.

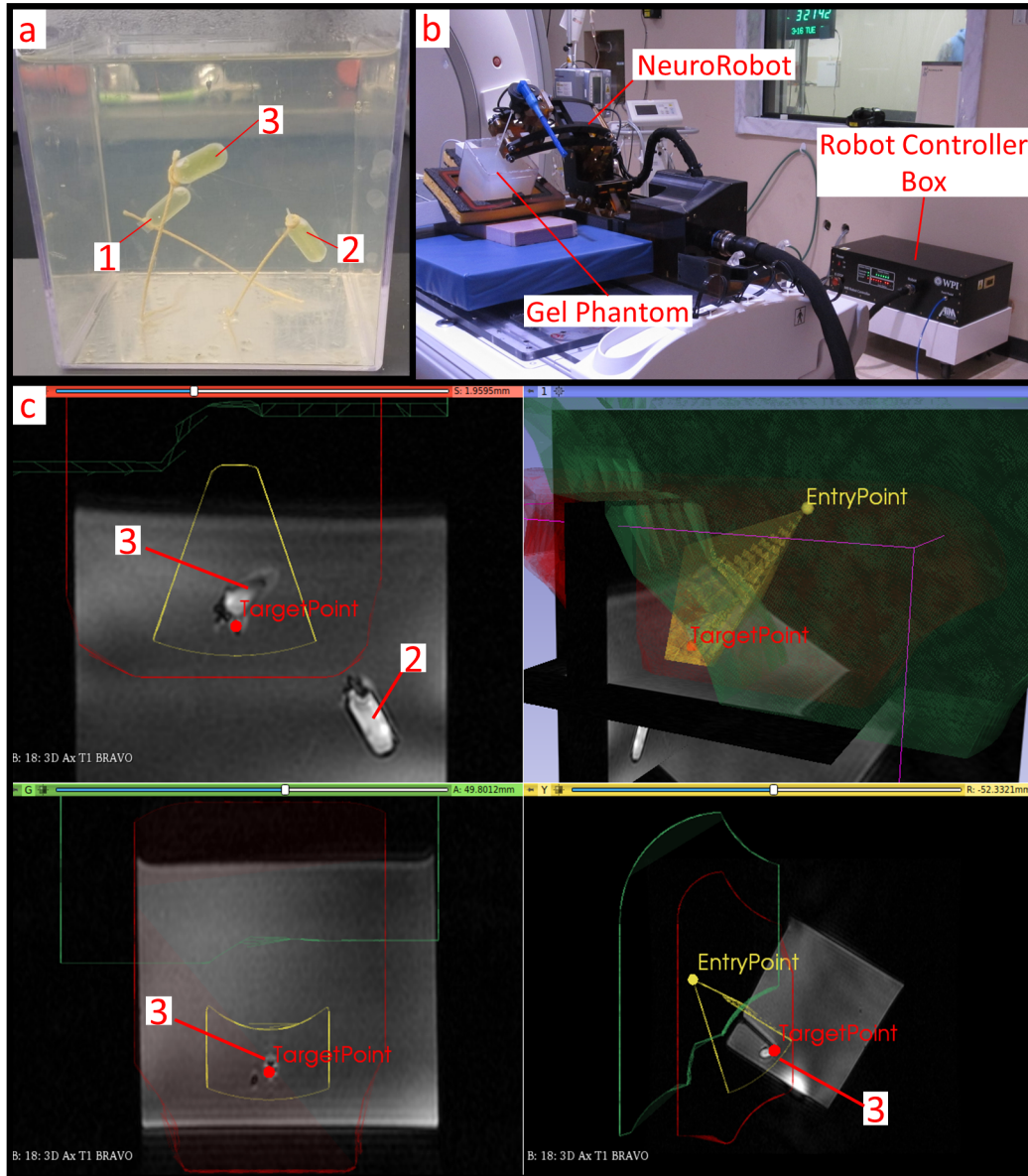


Figure 4.1: Phantom targeting test using the surgical planning toolkit. a) Gel-phantom with three randomly embedded MRI markers, b) Testing setup, c) A successful targetting of marker 3 using the Surgical Planning Toolkit. A TP was placed within the generated sub-workspace (shown in yellow) close to marker 3. The robot was then used to target the point and a validation scan was acquired.

Three MRI-compatible markers were randomly placed in a gel-phantom (figure

4.1a). During the experiment, the first attempt at placing the robot and targeting a marker deemed an insufficient TRW requiring the robot to be re-positioned for better reach. In the second attempt, marker 3 (figure 4.1c) was within the bounds of the TRW. The EPW was generated and an EP was placed along with generating the corresponding SW (figure 4.1c). A TP was placed close to marker 3 and the robot was targeted using that point. The probe was inserted and a validation scan was performed.

### 4.1.2 Results and Discussion

The results from the validation scan showed that the probe's actual tip slightly deviated from the pre-operative plan performed by the surgical planning toolkit with a RMS error of  $0.55mm$ . These values match the observed RMS error from the NeuroRobot's validation experiments conducted by Gandomi (Table 2 [96]). The error could be attributed to inconsistent *Z-Frame* registration through the registration sequencing, inaccuracies in the robot construction, and the straightness of the attached probe as well. However, Gandomi proved that the performance of the system can overcome the observed error through the NeuroRobot's control software.

## 4.2 Validating EP Identification

The surgical planning toolkit offers a semi-automated tool to help identify the burr-hole from a MRI acquisition. An experiment was carried out to test the accuracy of assisted burr-hole segmentation and center identification.

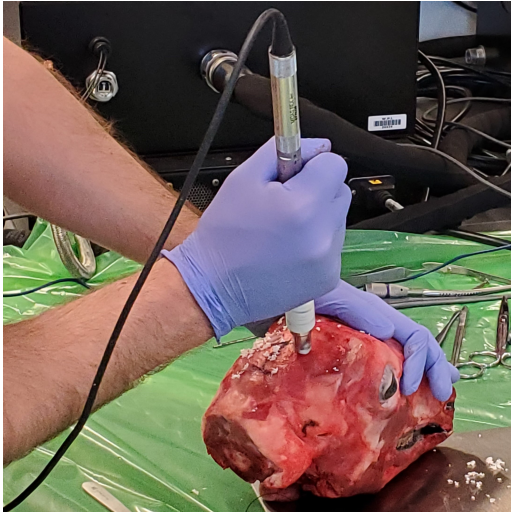
### 4.2.1 Procedure

The study was conducted on severed heads of sheep. Bi-lateral burr-holes were drilled on three sheep heads and MR/CT-compatible markers were stitched on the scalp. These MR/CT markers were used for mutual registration between MR and CT images. This experiment's goal was to validate the accuracy of burr-hole segmentation by comparing the identified center with the center of the burr-hole obtained from a CT scan. The CT scan offers a high fidelity image of the edges of the burr-hole in the cranium. The process listed below was conducted for each sheep head.

- 1) Prepare the sheep head and drill bilateral burr-holes in the cranium (figure 4.2a), and stitch MR/CT markers on the scalp (figure 4.2b).
- 2) Place the head on the CT scanner bed and run a scan.
- 3) Register the robot in the imaging frame.
  - 3.1) Place the robot on the MRI scanner bed and perform a registration scan (figure 4.3a).
  - 3.2) Obtain the registration matrix for the robot by performing a Z-Frame registration [83].
- 4) Place the head on the MR Scanner next to the robot (figure 4.3b).
- 5) Obtain a scan of the sheep head, push the DICOM images to the computer.
- 6) Identify and place the EP.
  - 6.1) Place markers around the boundary of the burr-hole and run the burr-hole segmentation.

- 6.2) Check if the identified EP is valid, if not modify the marker positioning from the previous step and re-run the segmentation (5.1).
- 7) Register the MR image with the CT image using the MR/CT markers for reference. (figure 4.4, 4.6, 4.5)
- 8) Validation of segmentations and detected centroids.
  - 8.1) Segment the burr-hole identified from the CT scan and compare the volume with the segmented region detected by the surgical planning toolkit on the MR images.
  - 8.2) Identify the centers of the burr-holes from the CT image in RAS coordinates and compare the values with that identified using the surgical planning toolkit.

Three sheep heads were used for this study. Bilateral cranial burr-holes were drilled (figure 4.2a) in each sheep head to increase the number of validation studies that could be performed. 3 MR/CT compatible markers were stitched on each head in a non co-linear pattern to allow for better registration (figure 4.2b). The registration scan of the robot was a T1 FLAIR sequence, and the imaging sequence was a 3D Ax T1 BRAVO sequence. The MR images were pushed to the workstation running the surgical planning toolkit using a DICOM push from the main MRI workstation.



(a) Burr-hole being drilled.



(b) Markers stitched on the scalp.

Figure 4.2: Preparing the sheep head for the study.





(a) Setting up the NeuroRobot for the procedure.



(b) Placing the sheep head next to the NeuroRobot.

Figure 4.3: Setting up the NeuroRobot and Sheep head for the procedure.

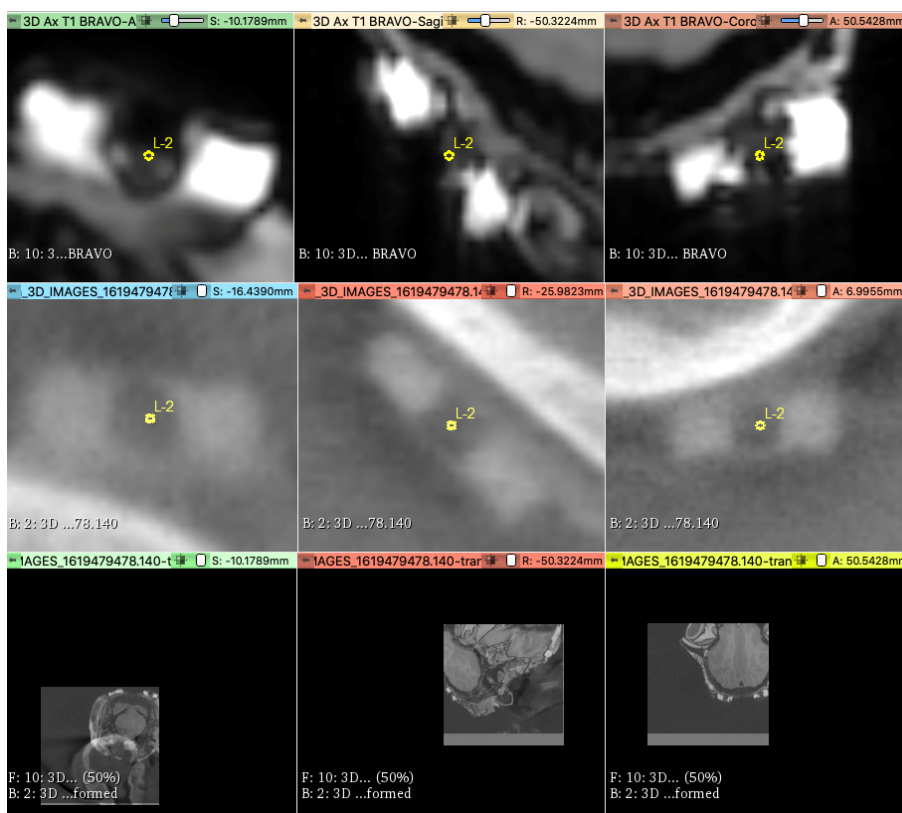


Figure 4.4: Placing Slicer markers where the MR/CT marker centers are visible. The top row is a view of the markers in the MR image, the middle row is a view of the markers in the CT image. The bottom row just shows the png of both the CT and MR before registration.

## 4.2.2 Results and Discussion

The burr-hole centers obtained from the CT images serve as the ground truth (figure 4.8), the accuracy of the assisted segmentation tool will be evaluated by calculating the RMS error between the identified center and ground truth. Table 4.1 shows the distance between CT (ground truth) and MR assisted identified locations of the center of each burr-hole in RAS coordinates from the registered MR-CT scans.



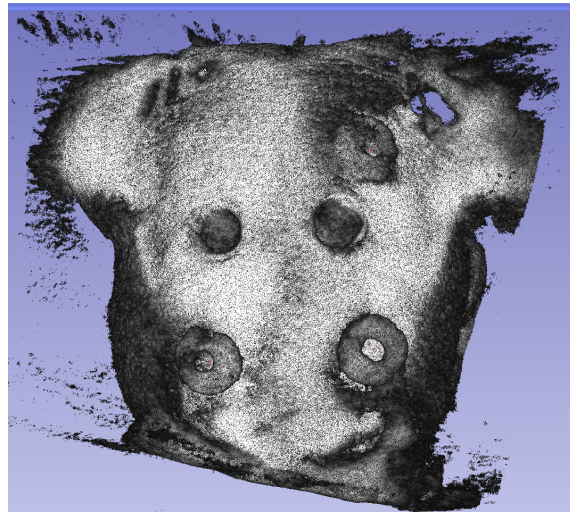


Figure 4.5: 3D rendered volume of the registered scan. Notice how prominent the features of the burr-hole and MR/CT markers are. The red markers are the ones placed in the image before (figure 4.4).

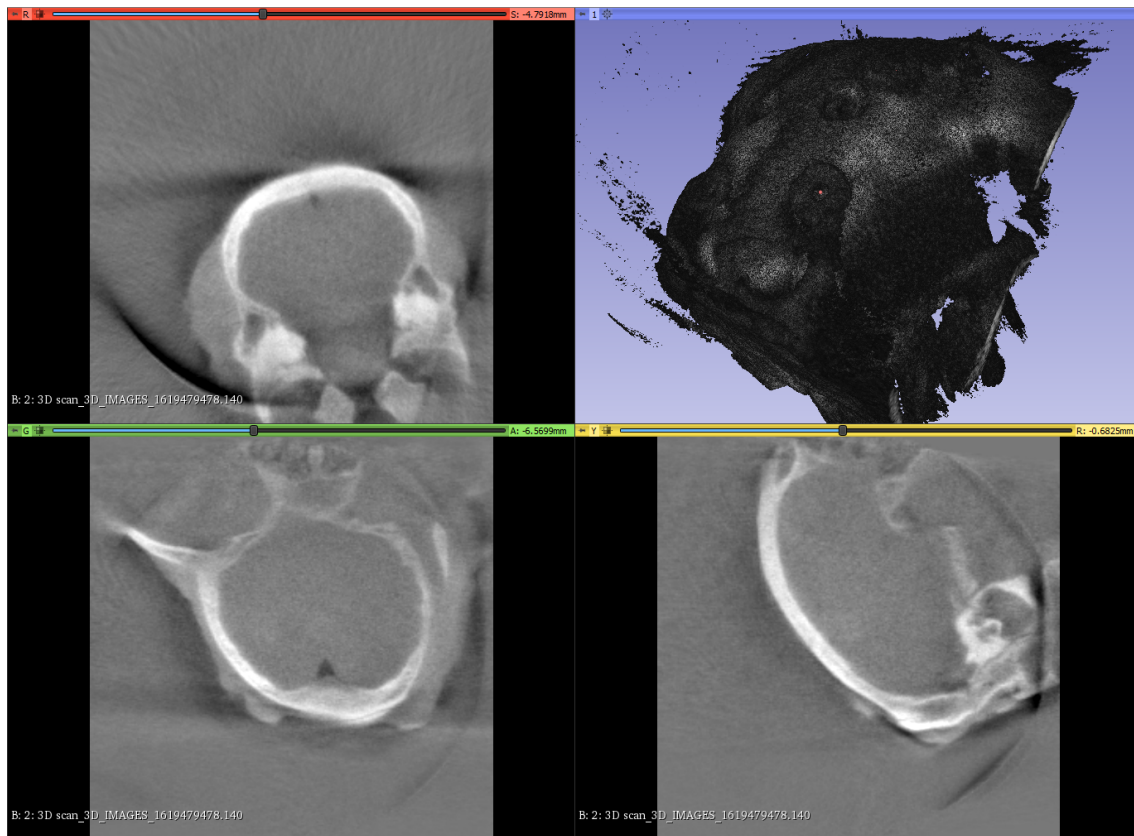


Figure 4.6: RAS planes of registered CT and MR scans along with 3D rendered view of the registered volumes.

Sub	Pos	Right (mm)			Anterior (mm)			Superior (mm)			ME (mm)
		G	A	E	G	A	E	G	A	E	
S1	L	-33.398	-32.998	-0.4	78.093	77.893	0.2	29.856	29.746	0.11	0.460
	R	-7.844	-7.964	0.12	73.417	73.337	0.08	26.773	26.693	0.08	0.164
S2	L	-43.406	-43.506	0.1	62.475	62.385	0.09	13.23	13.34	-0.11	0.174
	R	-17.973	-17.773	-0.2	62.24	62.06	0.18	11.524	11.734	-0.21	0.341
S3	L	-49.239	-49.109	-0.13	75.178	73.758	1.42	29.877	28.717	1.16	1.838
	R	-18.162	-18.102	-0.06	77.994	76.824	1.17	27.891	27.361	0.53	1.286

Table 4.1: Ground-truth, and Assisted estimate locations of burr-hole centers in RAS coordinates for each sheep subject. **G** - Ground-truth from CT Image, **A** - Assisted estimate from toolkit on MR image, **E** - Error (deviation) between ground-truth and assisted estimate using the toolkit, **ME** - Magnitude Error in mm, **S1** - Sheep 1, **S2** - Sheep 2, **S3** - Sheep 3, **Sub** - Subject, **Pos** - Burr-hole position, **L** - Left, **R** - Right.

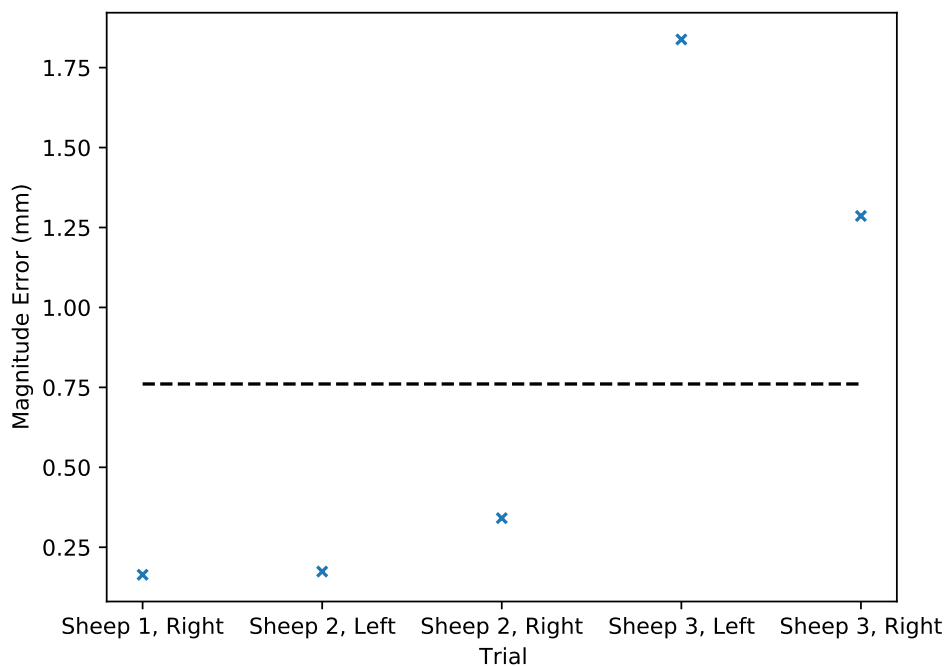


Figure 4.7: A plot of all the magnitude errors for each trial.

The mean deviation for all trials was  $(-0.095mm, 0.522mm, \text{ and } 0.26mm)$  in RAS coordinates respectively and the magnitude of the mean deviation for all trials  $0.7606mm$  (figure 4.7). The sub-millimeter deviation of the EPs identified by the

surgical planning toolkit helps develop confidence in this approach. However, it should be noted that the repeatability of this accuracy is solely dependent on how well the burr-hole is segmented by the extension which can be adversely affected by common characteristics such as noise, poor extreme point placements, etc.

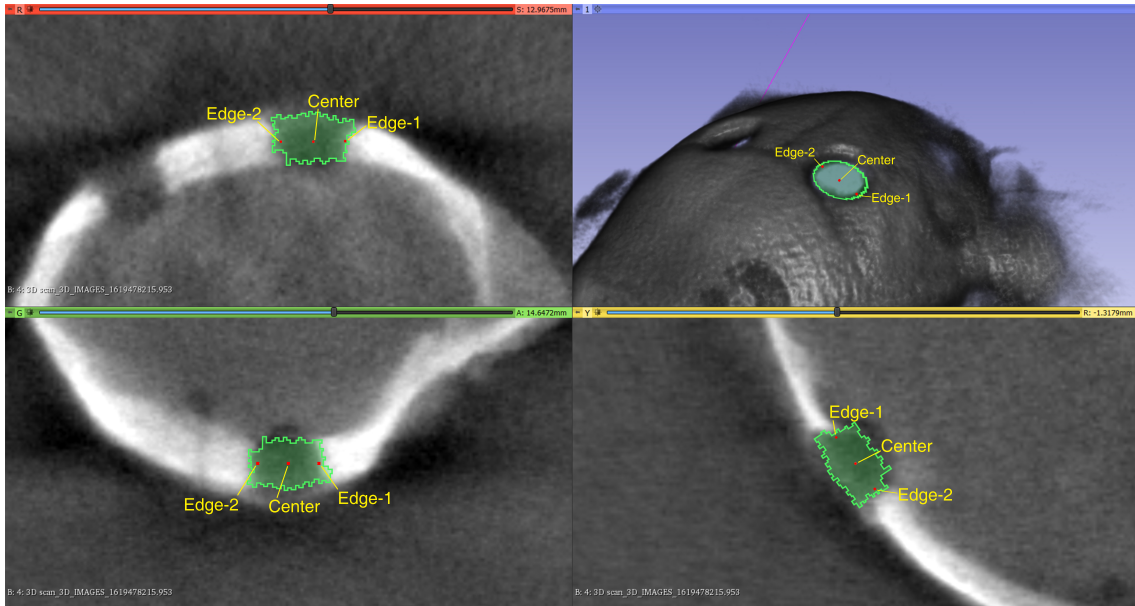


Figure 4.8: A segmented burr-hole from the first sheep head which is used as ground-truth. The segmentation was created on the CT volume after it had been registered with the MR volume.

In addition, the volume of the segmented regions of the burr-hole from the CT images (ground-truth) were compared with the volume of the segmented regions of the burr-hole from the MR images (assisted segmentation using the surgical planning toolkit). Table 4.2 shows the volumes of the segmented regions from both ground-truth from CT images, and assisted segments from MR images of the sheep. The mean error from all trials is  $2.739mm^3$ . On average, the segments created by the surgical planning toolkit on MR images are smaller than the segments created in the CT images, which is supported by the fact that the MR image is unable to visualize bone, loose tissue, water and other artifacts which reduce the visual region of the

burr-hole.

<b>Sub</b>	<b>Pos</b>	<b>Ground-truth Volume (<math>mm^3</math>)</b>	<b>Assisted Segment Volume (<math>mm^3</math>)</b>	<b>Error (<math>mm^3</math>)</b>
<b>S1</b>	<b>L</b>	75.751	71.441	4.31
	<b>R</b>	69.201	75.131	-5.93
<b>S2</b>	<b>L</b>	72.131	74.251	-2.12
	<b>R</b>	77.918	71.158	6.76
<b>S3</b>	<b>L</b>	75.43	66.339	9.091
	<b>R</b>	73.61	69.29	4.32

Table 4.2: Ground-truth, and Assisted estimate segments of burr-holes in  $mm^3$  for each sheep subject. **E** - Error (deviation) between ground-truth and assisted estimate, **S1** - Sheep 1, **S2** - Sheep 2, **S3** - Sheep 3, **Sub** - Subject, **Pos** - Burr-hole, **L** - left, **R** - right.

### 4.3 User study for EP identification

In addition to validating the accuracy of the burr-hole segmentation and EP identification, a user study was conducted to evaluate the surgeons preference with assisted EP identification over manual EP identification. Before the EP accuracy validation study was conducted (section 4.2), the accuracy of the assisted EP identification tool was qualitative and was based on the surgeon’s comfort with the tool. Hence, the goal of this study was to assign a metric to this qualitative accuracy estimate which was achieved by calculating the deviation of the EP’s position detected using the assisted EP identification tool with the position of the EP manually placed by the surgeon. In addition to obtaining the deviation, the surgeon was also asked to rate the resulting EP as unsatisfactory (P), satisfactory (A) or more than satisfactory (V) and the results of this study were collated.

### 4.3.1 Procedure

An MR image dataset was collated using scans from 6 previously completed ablation studies conducted on pigs. All scans were taken after drilling the burr-holes of which 2 images had bilateral burr-holes and the remaining 4 images had unilateral burr-holes. The images were all acquired using a 3T GE Signa™Architect™scanner and the acquisition sequences were all T1 BRAVO sequences. An neurosurgery resident was asked to identify the optimal EP for each burr-hole with and without assistance of the surgical planning toolkit. For each burr-hole, the expert would first identify and mark the EP without assistance and the RAS coordinates, and time taken to mark the EP were recorded after which the expert would identify and mark the EP with assistance of the surgical planning toolkit and the RAS (figure 4.9), and time taken to identify the EP were recorded. The expert would then qualitatively evaluate the accuracy of the segmented burr-hole and the identified EP. If the results were unsatisfactory the expert would perform the assisted identification again and qualitatively evaluate the result, otherwise the expert would continue with the next burr-hole.

### 4.3.2 Results and Discussion

	<i>Right (mm)</i>	<i>Anterior (mm)</i>	<i>Superior (mm)</i>	Assisted Time(s)	Un-Assisted Time(s)
Average	1.2	1.21	1.31	20.74	8.3
Min	0.147	0.01	0.0	8	4.2
Max	2.99	3.51	3.42	29.4	16.4

Table 4.3: Comparing the absolute difference between EP placed by the expert with and without assistance for all successful trials.

	<i>Right (mm)</i>	<i>Anterior (mm)</i>	<i>Superior (mm)</i>
Average	0.96	0.67	0.58
Min	0.04	0.0	0.0
Max	2.03	2.45	2.46

Table 4.4: Comparing the absolute error between assisted EP identification over two trials on the same burr-hole. The first trial was unsatisfactory to the surgeon, the second trial was more than satisfactory.

Table 4.3 shows the absolute difference between the EPs placed by the expert with and without assistance of the surgical planning toolkit. Average deviations of  $1.20mm$ ,  $1.21mm$ , and  $1.31mm$  in RAS coordinates respectively were observed, with an average of 20.74 seconds for all trials which were performed with the assistance of the surgical planning toolkit. In contrast, the average time taken to select the EP without assistance turned out to be 8.3 seconds. While the time taken to manually identify the EPs was shorter on average as compared with assisted identification, after discussing the outcome of the results with the expert, the assisted EPs were more than satisfactory and deemed better than those placed without assistance. This conclusion was solely drawn from the experts opinion with no source of ground truth. The experiment mentioned in Section 4.2 was implemented after this user study to evaluate the accuracy of the EP identification using assistance from the surgical planning toolkit.

Two trials yielded unsatisfactory segmentation and EPs due to high noise in the MR images. For the first case the trial was conducted twice over, and once over for the second case. We calculated the absolute error between EPs which were deemed unsatisfactory with the EPs placed without assistance and obtained an average value of  $0.98mm$ ,  $1.9mm$ ,  $1.08mm$  in RAS coordinates respectively.

Table 4.4 describes the repeatability in the identified EP using the surgical planning toolkit between both trials for each burr-hole experiment. The average devi-

ations between the first and second trials are less than 1mm, with the RAS values being  $0.96mm$ ,  $0.67mm$ , and  $0.58mm$  respectively.

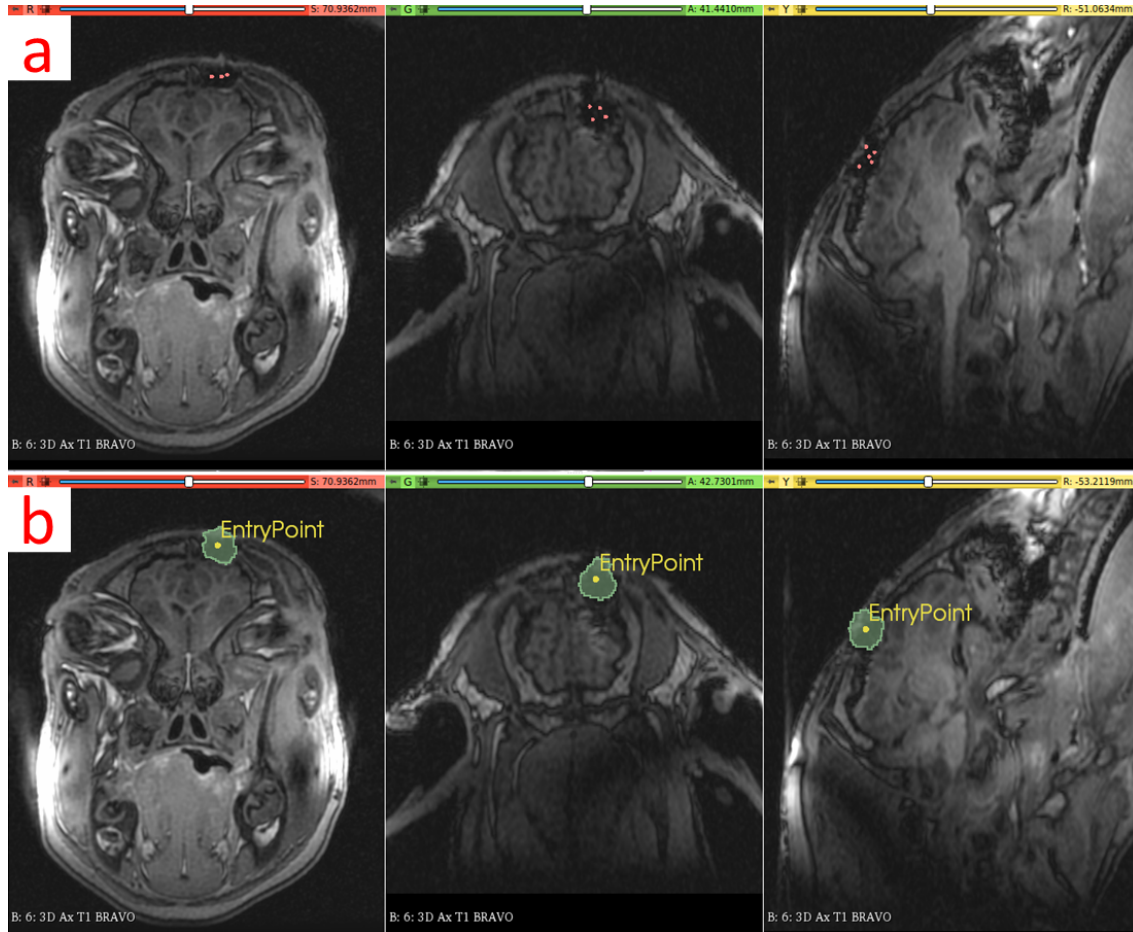


Figure 4.9: Assisted segmentation of the burr-hole region. a) 8 markups placed inside the burr-hole region to annotate the extremum. b) Result of the assisted segmentation using the surgical planning toolkit, along with EP identification.

## 4.4 Pig trials with modified workflow

Finally, the surgical planning toolkit was tested on subacute pig ablation experiments to evaluate its applicability to ablation studies. A modified workflow was established which serves as the defacto workflow for further animal studies.

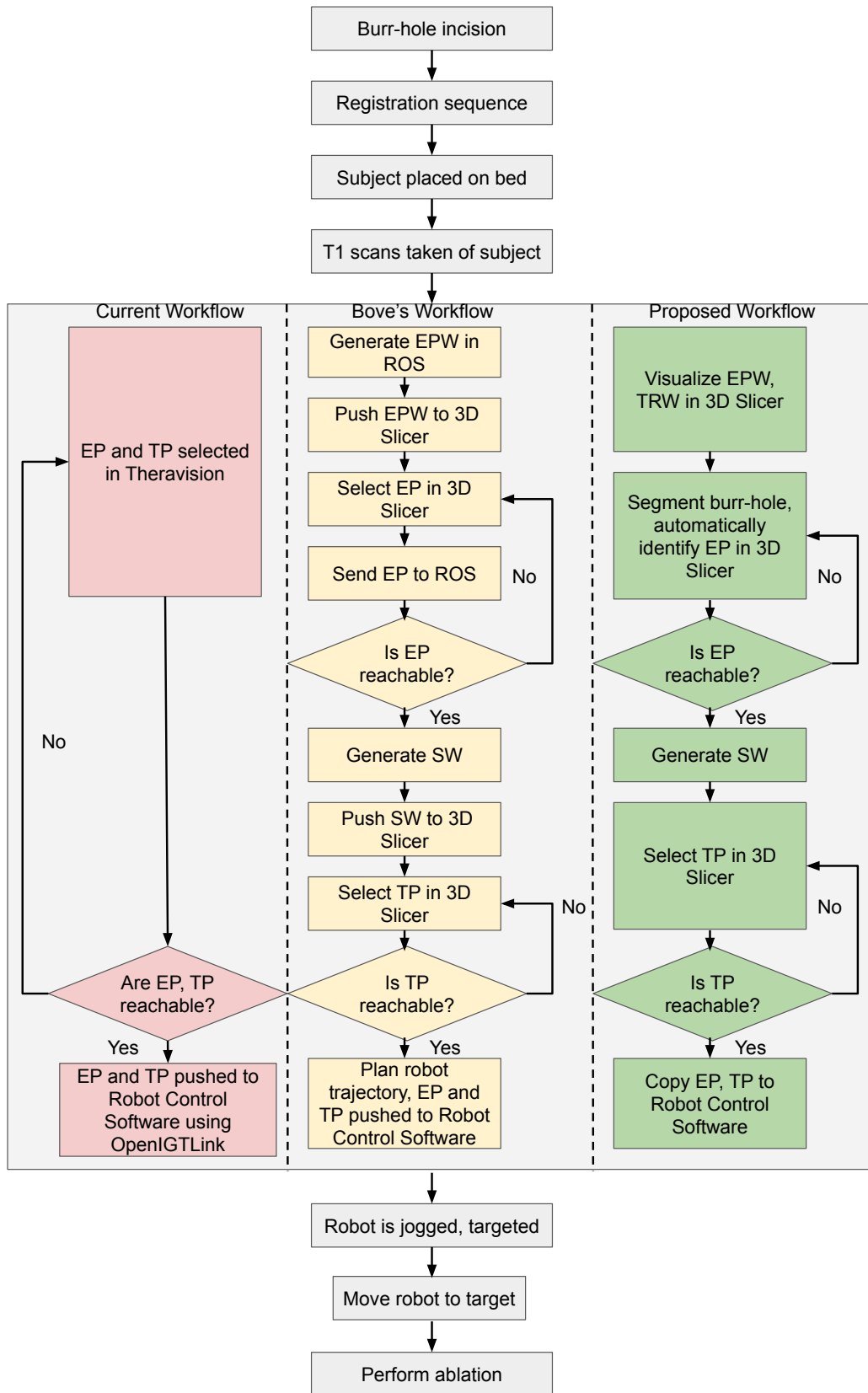




Figure 4.10: The proposed workflow as performed using the surgical planning toolkit. The current workflow (red) and Bove’s[33] workflow (yellow) are shown beside the proposed workflow (green) for comparison.

### 4.4.1 Modified Workflow

- (Optional) Perform a pre-operative scan to identify an optimal location to drill the burr-hole to reach the desired treatment location.
- 1) Prepare the subject and drill a burr-hole in the cranium.
  - 2) Register the robot in the imaging frame.
    - 3.1) Place the robot on the MRI scanner bed and perform a registration scan.
    - 3.2) Obtain the registration matrix for the robot by performing a Z-Frame registration [83].
  - 3) Place the subject on the MR Scanner next to the robot, and acquire an image of the subject.
  - 4) Generate and visualize workspaces.
    - 4.1) Specify the probe Specifications.
    - 4.2) Generate the TRW and EPW and augment them on the 2D slices and 3D volume render of the subjects head.
    - 4.3) Check if the treatment location lies within the TRW.
    - 4.4) If the treatment lies outside the TRW, reposition the robot and repeat from step 2), else continue.
  - 5) Identify and place the EP.

- 5.1) Place markers around the boundary of the burr-hole and run the burr-hole segmentation.
- 5.2) Check if the identified EP is valid, if not modify the marker positioning from the previous step and re-run the segmentation (5.1).
- 6) Generate the SW given the identified EP.
- 7) Place the TP on or closest to the desired treatment location.
- 8) Send the final EP and TP positions to the robot.
- 9) Insert the probe and continue with the ablation experiments.

Figure 4.10 shows how the proposed workflow differs from the current workflow and Bove’s workflow [33]. The major differences between the proposed workflow and the current workflow are the addition of intermediate steps while identifying the EP and TP. Once the robot is registered in the imaging plane and a pre-op scan has been taken of the subject, the surgical planning toolkit renders the EPW, and TRW directly in 3D Slicer instead of generating the same in ROS (from Bove’s workflow). The surgeon is then asked to place markers to annotate the extremum of the burr-hole and the burr-hole is segmented and the EP is automatically placed at the center of the segment. If the resulting EP is unsatisfactory or the EP is not reachable, a prompt asks the surgeon to move the EP. Once the EP is placed and is reachable, the SW is rendered and visualized in 3D Slicer. The surgeon then selects a TP within the bounds of the SW. If the selected treatment lies outside the SW, a prompt indicates that the TP is not reachable. Finally, the EP, and TP are copied to the Robot Control Software and the robot is jogged to the target the treatment for the ablation procedure. [97] shows a recorded video of the entire workflow on a previously conducted pig trial.

# Chapter 5

## Conclusion

A surgical planning toolkit along with its underlying components was developed for a NeuroRobot that performs stereotactic surgical interventions.

### 5.1 Key Contributions

#### 5.1.1 Structured surgical workflow design

A new surgical workflow was presented in section 4.4 which should substantially reduce the time taken to prepare the NeuroRobot and subject for the surgical intervention. The workflow was validated by testing it on 6 prior pig trials. Modifications to the current workflow include:

- 1) Robot specific workspaces (TRW, EPW, and SW) are augmented on the pre-operative MR images of the subject. These inform the surgeon of the robots' reach for the surgical task and whether or not the robot should be re-positioned earlier into the surgical workflow as compared to the current workflow.
- 2) Assisted burr-hole segmentation and EP identification replace the manual

steps of identifying the burr-hole undertaken by the surgeon in the current workflow while offering a more accurate EP detection. This allows for safer, collision-free probe insertions.

### 5.1.2 Intuitive toolkit design

The toolkit was designed as a Slicer module and incorporates the MVC design pattern like Slicer. An interactive GUI is presented which is structured like a stateflow. Functionalities offered by the toolkit are disabled by default and are only enabled when certain steps are performed. For example, The steps to generate a TRW and EPW require an MR Image to be loaded, and the registration matrix and probe specifications to be entered into their respective input fields. Similarly, generating the SW requires an EP to be identified which in turn needs the TRW to be generated to check if its within the robot's reach. The strict adherence to maintaining the chronological order of the workflow help make this toolkit more suitable for subject studies.

**Model** The Model from the MVC architecture is analogous to the toolkit's MRML node. The MRML node stores the state of the toolkit and offers convenience handlers to read and write from a file to save the active scene. The methods to implement this were presented and an overview of the scene's information that is saved was discussed.

**Logic** The toolkit's Logic implements a group of algorithms that serve as algorithmic handlers for the interactive events triggered by the GUI. It includes endpoints to generate workspaces for the TRW, EPW, and SW. Algorithms to visualize the workspaces were presented and the more efficient technique using Meshlab was implemented in the Logic. The Logic also includes algorithms to segment

burr-holes and identify the segmented region's centroid which serves as the EP. An overview of classical segmentation algorithms was presented along with a DL based technique of which the DL based algorithm was chosen for its accuracy.

MRML All the data that is generated by the toolkit can be saved in a Slicer scene. This allows the surgeon to re-load the same scene that was used during a surgery and inspect the accuracy of the tools post-op. Data stored can also be used for further training of the assisted burr-hole segmentation.

### **5.1.3 Experimentation and Validation of the toolkit's components**

Each component of the toolkit was validated for accuracy. Four experiments were carried out for this purpose,

- 1) The first experiment validated the accuracy of the workspace generation and visualization through a gel-phantom study. The resulting accuracy was  $0.55mm$  which was observed through a validation scan at the end of the experiment.
- 2) The second experiment validated the accuracy of burr-hole identification using the toolkit. The experiment was carried out on 3 sheep heads with bi-lateral burr-holes on each sheep. Both a CT and MR image were captured for each sheep and the exact location of the EP was identified using the CT images while the assisted segmentation and EP identification tool calculated the EP from the MR images. Both the CT and MR image were registered using MT/CT-compatible markers and the final deviation between the exact location and estimated location of the EP was calculated in RAS coordinates.

The resulting deviations were  $(-0.095mm, 0.522mm, \text{ and } 0.26mm)$  in RAS coordinates respectively.

- 3) The third experiment conducted was aimed at identifying a metric of qualitative accuracy from the surgeon's opinion. A dataset of prior pig study experiments was collated and the surgeon was tasked with identifying the EP for each MR image with and without assistance of the surgical planning tool. The average deviation between the EPs identified with and without assistance was  $(1.20mm, 1.21mm, 1.31mm)$  in RAS coordinates while the time taken to identify the EP averaged to 20.74 seconds. The surgeon's opinion was taken after each experiment and the conclusion drawn from the study was that the assistance of the surgical planning toolkit proved more beneficial than manually placing the EP.
- 4) The fourth and last experiment tested the entire surgical planning toolkit through a subacute pig ablation experiment. The surgeon was able to successfully carry out the experiment using the toolkit and a modified workflow was devised for future experiments.

## 5.2 Impact of COVID on this work

The COVID-19 pandemic brought the entire world to a standstill from March 2020, through June 2021. Since I took up this project after the lockdown had begun, the deliverables for this project were designed to be tasks that could be completed remotely. While the toolkit was designed and developed entirely remotely barring a few opportunities to use the toolkit in a clinical pig study, I was unable to expand on the user study and have more than one user test the toolkit. The user testing the toolkit was a member of the team working on developing this toolkit while being

a resident neurosurgeon. His knowledge of the toolkit and expertise in the domain ranks him as a baseline for what an *expert* user would be for future data points. To further development, the same procedure as conducted in 4.3 should be performed with several candidate users to assess the repeatability, accessibility, and time taken to identify EPs and perform the targeting of the NeuroRobot.

## 5.3 Future Work

The proposed surgical planning toolkit was successfully implemented and validated for accuracy. However, there remain some features that could enhance this toolkit. The following sections expand on these enhancements and help scope the future development of this extension.

### 5.3.1 Feature: Entry Point Region Identification

The current method to detect the EP identifies the center of the segmented burr-hole as the EP. While this technique works well and helps generate a sub-workspace to reach the treatment, the EP identification shouldn't be limited to a single point instead a region of possible EPs should be identified allowing the surgeon to move the EP around to obtain a larger sub-workspace to cover more treatment locations.

Figure 5.1 shows an ROI of a region which lies within the burr-hole. The dimensions of the region match the cross-sectional area of the drill-bit used to drill the burr-hole. The cross-sectional radius of the drill bit can be used to draw the circumference of the region with the center of the drill bit aligned with the detected center of the segmented burr-hole.

## CHAPTER 5. CONCLUSION

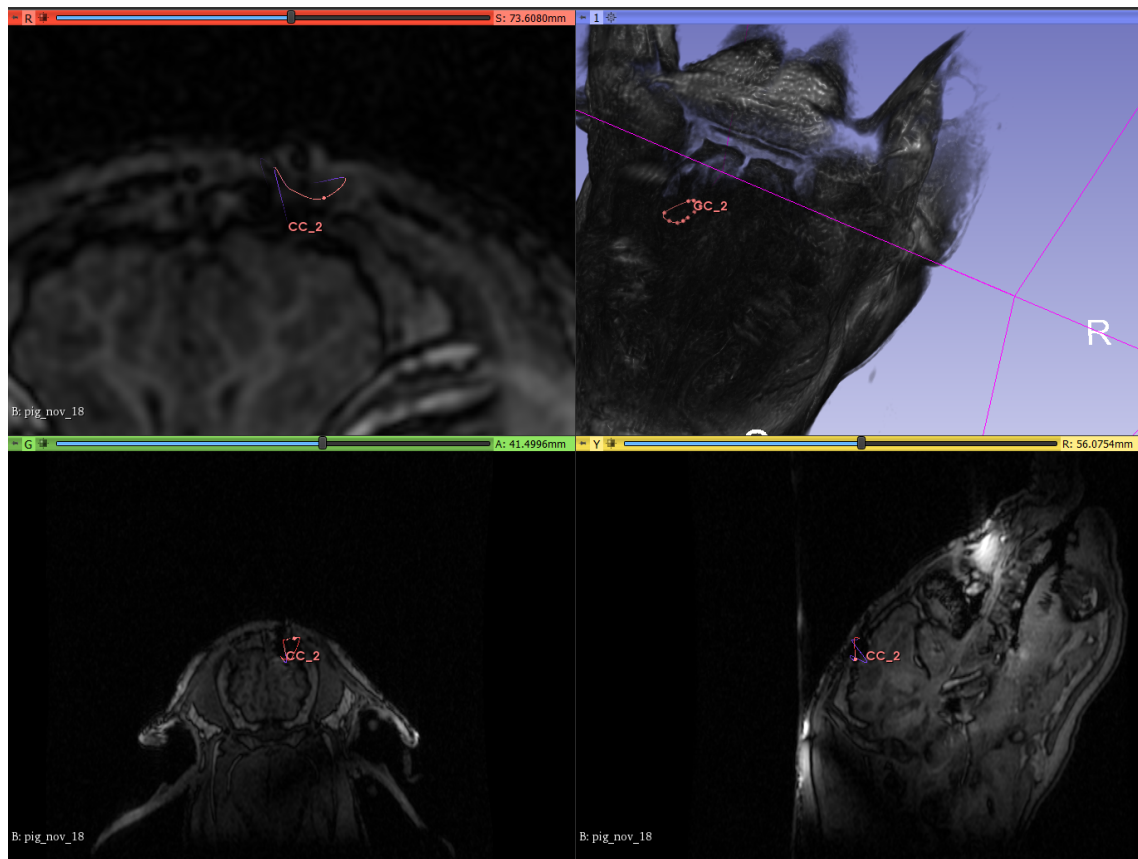


Figure 5.1: A rendering of an ROI showing the cross-sectional area of the drill bit.

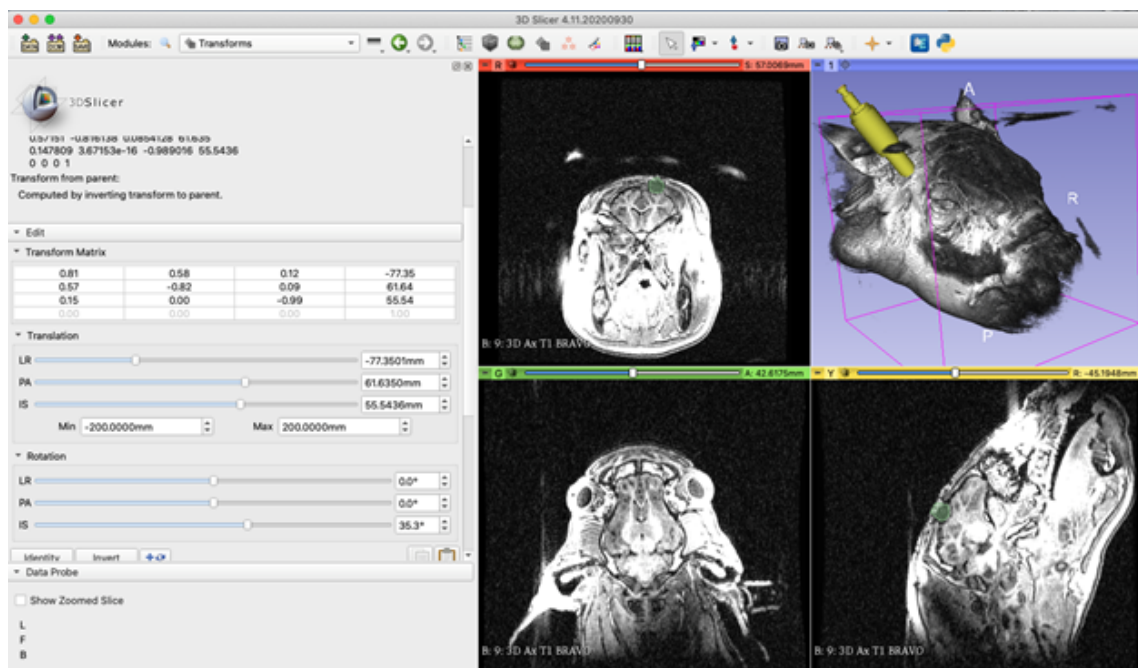


Figure 5.2: A 3D CAD imported mesh of the drill bit aligned with the insertion angle of the burr-hole.



The next step is to identify the axis of insertion (drill bit angle of approach) of the drill bit while drilling the burr-hole as this will help align the cross-sectional region to obtain the largest possible Entry Point Region (figure 5.2). It could be possible to use the principle axes of the segmented burr-hole to inform the identification of this axis of insertion. Creating this feature as a semi-assisted feature might be most beneficial with the final step of correcting the axis of insertion to the surgeon.

The benefits of using this tool would enhance the generated sub-workspace, allowing for edge cases where a steep angle is required for the probe insertion to reach the actual treatment location.

### **5.3.2 Feature: Enabling communication between the Robot Control Software and Surgical Planning Toolkit**

The current extension doesn't support sending the selected EP and TP to the Robot Control Software. The Surgical Planning Toolkit could support communicating over OpenIGTLink by setting up the toolkit as a Client and pushing the EP and TP coordinates as a `Position` message to the Robot Control Software which is setup as a Server.

### **5.3.3 Feature: Motion Planning for collision avoidance**

The research presented by Bove [33] largely impacted this work. While the toolkit in its current state attempts to assist the surgeon is pre-operative surgical planning, it still requires scrutiny to enhance the safety features of which collision avoidance while targeting the arm is an important one. Implementing motion planning will not only improve the safety of the robot but also reduce the time taken to make the robot reach the desired location to perform the intervention. Similar to how

the EP and TP are shared using OpenIGTLink, the motion plan generated can be shared with the Robot Control Software as a Traj message for the trajectory. Bove implemented motion planning (figure 5.3) using ROS, MoveIt, and OMPL. However, a similar approach might not be possible to use in the surgical planning toolkit as all tools are directly integrated into 3D Slicer. The first step to perform motion planning would involve computing the MR volume of the subject as a collision mesh. Similarly, the environment (scanner bore hole) and robot itself should be computed as a collision mesh. Once computed, and the kinematics of the same are known, traditional motion planning approaches can be applied.

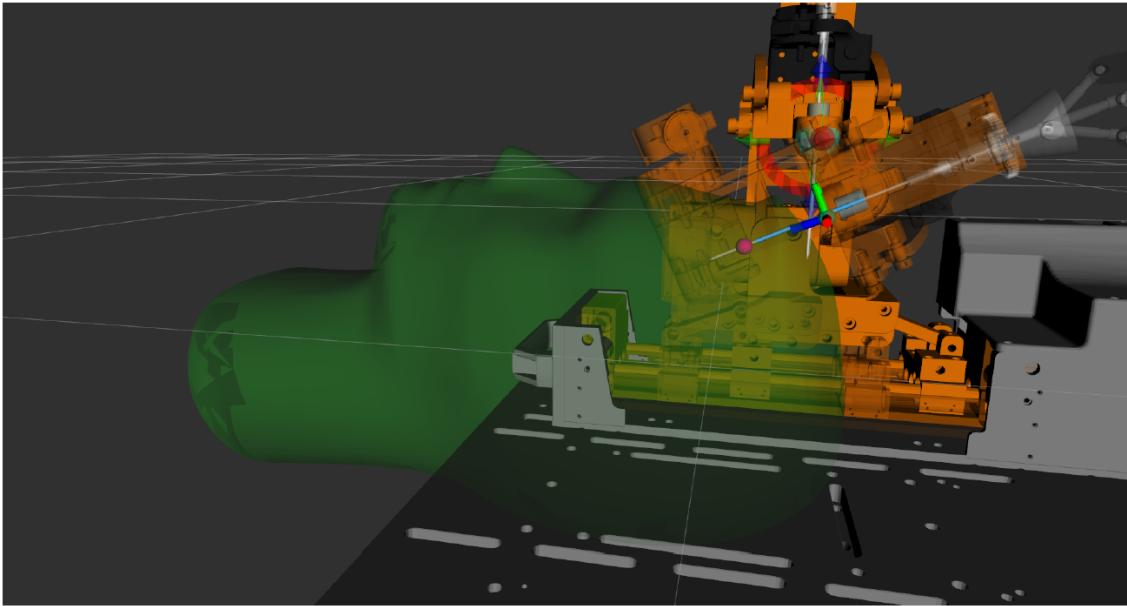


Figure 5.3: The NeuroRobot loaded in RViz with markers for waypoints generated by the motion planning service in Bove’s dissertation [33]

### 5.3.4 Feature: Surgical Planning during insertion

In addition to performing motion planning for collision avoidance, segmenting and identifying critical tissues and arteries/lobes and including these tissues as collisions for the motion planner would serve as an additional constraint to the sub-workspace

generation and TP selection algorithms. This additional information would add to the safety of using such a system during clinical trials. Similar work has been attempted in [20] (figure 5.4) where the authors designed a Slicer extension for manual Deep Brain Stimulation (DBS) surgeries and surgically planned the intervention using pre-operative CT scans.

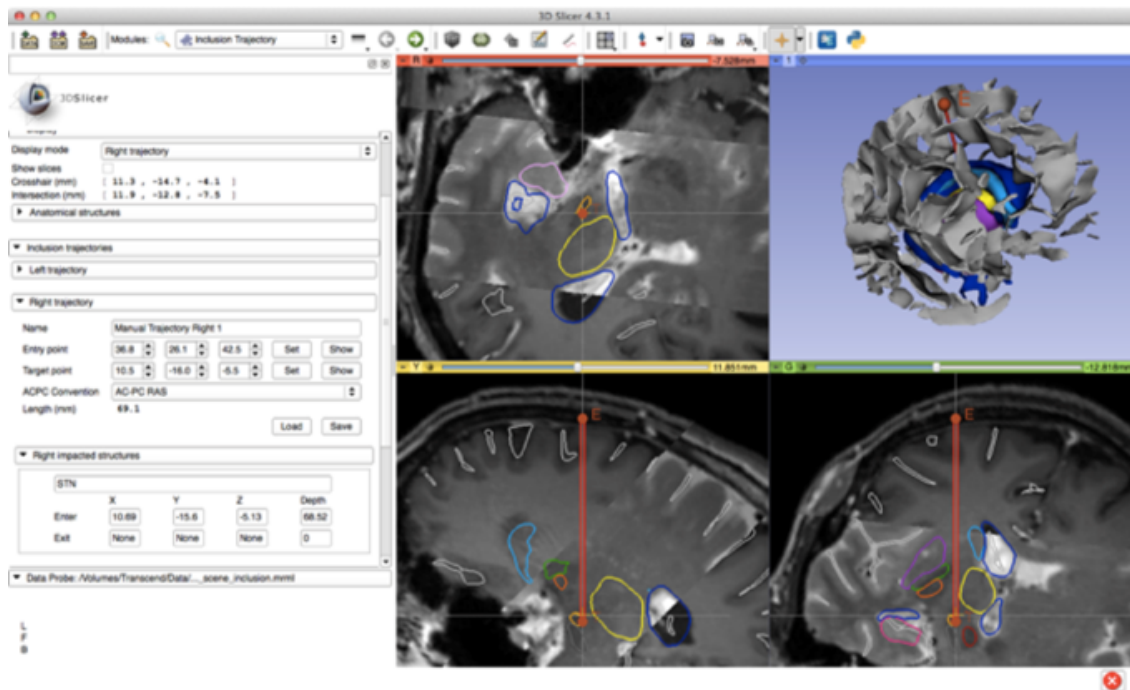


Figure 5.4: Surgical planning with collision avoidance using the PyDBS extension in Slicer [20]

### 5.3.5 Modifications to the current toolkit

#### 5.3.5.1 Burr-hole segmentation

Faster approaches to identify the EP of the burr-holes should be investigated. The use of the NVIDIA AIAA tool also poses problems with its dependency on Linux based CUDA enabled workstations or servers and it hinders the cross platform compatibility which is crucial to the modularity of the robot. While a possible

solution to this problem would be to establish a remote server to run the NVIDIA AIAA tool for segmentation and connect it to the 3D Slicer Extension for the surgical planning toolkit which could be run on any workstation (Windows, Mac, Linux), it might adversely affect the safety of the procedure and privacy of the patient by sharing medical records in real-time beyond the local network of the MR suite.

### **5.3.5.2 Workspace Visualization**

Meshlab server, the current software that is used in the toolkit to generate a mesh from the pointset of the workspaces, is an additional dependency for this toolkit. To make it more modular, other methods of generating meshes from a point-set should be investigated like the VCG library or openGL.

# Bibliography

- [1] K. Leibrandt, C. Bergeles, and G.-Z. Yang, “Concentric tube robots: Rapid, stable path-planning and guidance for surgical use,” *IEEE Robotics & Automation Magazine*, vol. 24, no. 2, pp. 42–53, 2017.
- [2] L. Joskowicz, R. Shamir, M. Freiman, M. Shoham, E. Zehavi, F. Umansky, and Y. Shoshan, “Image-guided system with miniature robot for precise positioning and targeting in keyhole neurosurgery,” *Computer Aided Surgery*, vol. 11, no. 4, pp. 181–193, 2006.
- [3] M. S. Lasser, M. Doscher, A. Keehn, V. Chernyak, E. Garfein, and R. Ghavamian, “Virtual surgical planning: A novel aid to robot-assisted laparoscopic partial nephrectomy,” *Journal of endourology*, vol. 26, no. 10, pp. 1372–1379, 2012.
- [4] P. Kazanzides, T. Xia, C. Baird, G. Jallo, K. Hayes, N. Nakajima, and N. Hata, “A cooperatively-controlled image guided robot system for skull base surgery,” *Studies in health technology and informatics*, vol. 132, pp. 198–203, 2008.
- [5] J. J. Xia, C. V. Phillips, J. Gateno, J. F. Teichgraeber, A. M. Christensen, M. J. Gliddon, J. J. Lemoine, and M. A. Liebschner, “Cost-effectiveness analysis for computer-aided surgical simulation in complex cranio-maxillofacial surgery,” *Journal of oral and maxillofacial surgery*, vol. 64, no. 12, pp. 1780–1784, 2006.

- [6] Jessica, *Most expensive medical procedures in the u.s., ranked*, Jul. 2020. [Online]. Available: <https://www.cbsnews.com/pictures/most-expensive-medical-procedures-without-insurance/48/>.
- [7] *Cranialmap 3.0*. [Online]. Available: <https://www.stryker.com/us/en/navigation/products/cranial-map-3-0.html>.
- [8] *Image-guided surgery products - stealthstation flexent*. [Online]. Available: <https://www.medtronic.com/us-en/healthcare-professionals/products/ear-nose-throat/image-guided-surgery/stealthstation-flexent.html>.
- [9] *Medical software and hardware innovators*, Jun. 2021. [Online]. Available: <https://www.brainlab.com/>.
- [10] L. Leksell, "Stereotactic radiosurgery," *Journal of Neurology, Neurosurgery & Psychiatry*, vol. 46, no. 9, pp. 797–803, 1983.
- [11] N. McDannold, G. T. Clement, P. Black, F. Jolesz, and K. Hynynen, "Transcranial magnetic resonance imaging-guided focused ultrasound surgery of brain tumors: Initial findings in 3 patients," *Neurosurgery*, vol. 66, no. 2, pp. 323–332, 2010.
- [12] J. Y. Kang, C. Wu, J. Tracy, M. Lorenzo, J. Evans, M. Nei, C. Skidmore, S. Mintzer, A. D. Sharan, and M. R. Sperling, "Laser interstitial thermal therapy for medically intractable mesial temporal lobe epilepsy," *Epilepsia*, vol. 57, no. 2, pp. 325–334, 2016.
- [13] R. J. McNichols, A. Gowda, M. Kangasniemi, J. A. Bankson, R. E. Price, and J. D. Hazle, "Mr thermometry-based feedback control of laser interstitial thermal therapy at 980 nm," *Lasers in Surgery and Medicine: The Official*

*Journal of the American Society for Laser Medicine and Surgery*, vol. 34, no. 1, pp. 48–55, 2004.

- [14] C. B. Ostertag, H. Mennel, and M. Kiessling, “Stereotactic biopsy of brain tumors,” *Surgical neurology*, vol. 14, no. 4, pp. 275–283, 1980.
- [15] N. Dorward, T. Paleologos, O. Alberti, and D. Thomas, “The advantages of frameless stereotactic biopsy over frame-based biopsy,” *British journal of neurosurgery*, vol. 16, no. 2, pp. 110–118, 2002.
- [16] J. J. Van Gompel, F. B. Meyer, W. R. Marsh, K. H. Lee, and G. A. Worrell, “Stereotactic electroencephalography with temporal grid and mesial temporal depth electrode coverage: Does technique of depth electrode placement affect outcome?” *Journal of neurosurgery*, vol. 113, no. 1, pp. 32–38, 2010.
- [17] I. Chang, “Finite element analysis of hepatic radiofrequency ablation probes using temperature-dependent electrical conductivity,” *Biomedical engineering online*, vol. 2, no. 1, pp. 1–18, 2003.
- [18] K. Cleary and T. M. Peters, “Image-guided interventions: Technology review and clinical applications,” *Annual review of biomedical engineering*, vol. 12, pp. 119–142, 2010.
- [19] P. R. DeLucia, R. D. Mather, J. A. Griswold, and S. Mitra, “Toward the improvement of image-guided interventions for minimally invasive surgery: Three factors that affect performance,” *Human factors*, vol. 48, no. 1, pp. 23–38, 2006.
- [20] T. D’Albis, C. Haegelen, C. Essert, S. Fernández-Vidal, F. Lalys, and P. Janin, “PyDBS: an automated image processing workflow for deep brain stimulation surgery,” *Int J Comput Assist Radiol Surg*, vol. 10, no. 2, pp. 117–128, Feb. 2015.

- [21] A. Fomenko and D. Serletis, “Robotic Stereotaxy in Cranial Neurosurgery: A Qualitative Systematic Review.,” *Neurosurgery*, vol. 83, no. 4, pp. 642–650, Oct. 2018.
- [22] T. R. K. Varma and P. Eldridge, “Use of the neuromate stereotactic robot in a frameless mode for functional neurosurgery,” *The International Journal of Medical Robotics and Computer Assisted Surgery*, vol. 2, no. 2, pp. 107–113, 2006. DOI: <https://doi.org/10.1002/rcs.88>. eprint: <https://onlinelibrary.wiley.com/doi/pdf/10.1002/rcs.88>. [Online]. Available: <https://onlinelibrary.wiley.com/doi/abs/10.1002/rcs.88>.
- [23] C. Dorfer, B. Rydenhag, G. Baltuch, V. Buch, J. Blount, R. Bollo, J. Gerrard, D. Nilsson, K. Roessler, J. Rutka, A. Sharan, D. Spencer, and A. Cukiert, “How technology is driving the landscape of epilepsy surgery,” *Epilepsia*, vol. 61, no. 5, pp. 841–855, Mar. 2020.
- [24] G. Cole, J. Pilitsis, and G. S. Fischer, “Design of a robotic system for mri-guided deep brain stimulation electrode placement,”
- [25] A. Kim, S. Tatter, G. Rao, S. Prabhu, C. Chen, P. Fecci, V. Chiang, K. Smith, B. Williams, A. Mohammadi, K. Judy, A. Sloan, Z. Tovar-Spinoza, J. Baumgartner, C. Hadjipanayis, and E. Leuthardt, “Laser Ablation of Abnormal Neurological Tissue Using Robotic NeuroBlate System (LAANTERN): 12-Month Outcomes and Quality of Life After Brain Tumor Ablation,” *Neurosurgery*, vol. 87, no. 3, E338–E346, Sep. 2020.
- [26] K. Gupta, A. Dickey, R. Hu, E. Faught, and J. Willie, “Robot Assisted MRI-Guided LITT of the Anterior, Lateral, and Medial Temporal Lobe for Temporal Lobe Epilepsy,” *Front Neurol*, vol. 11, no. 572334, Nov. 2020.



- [27] H. J. Marcus, V. N. Vakharia, S. Ourselin, J. Duncan, M. Tisdall, and K. Aquilina, "Robot-assisted stereotactic brain biopsy: Systematic review and bibliometric analysis," *Child's Nervous System*, vol. 34, no. 7, pp. 1299–1309, 2018.
- [28] J. Tokuda, G. S. Fischer, X. Papademetris, Z. Yaniv, L. Ibanez, P. Cheng, H. Liu, J. Blevins, J. Arata, A. J. Golby, T. Kapur, S. Pieper, E. C. Burdette, G. Fichtinger, C. M. Tempny, and N. Hata, "Openigtlink: An open network protocol for image-guided therapy environment," *The International Journal of Medical Robotics and Computer Assisted Surgery*, vol. 5, no. 4, pp. 423–434, 2009. DOI: <https://doi.org/10.1002/rcs.274>. eprint: <https://onlinelibrary.wiley.com/doi/pdf/10.1002/rcs.274>. [Online]. Available: <https://onlinelibrary.wiley.com/doi/abs/10.1002/rcs.274>.
- [29] C. J. Nycz, R. Gondokaryono, P. Carvalho, N. Patel, M. Wartenberg, J. G. Pilitsis, and G. S. Fischer, "Mechanical validation of an mri compatible stereotactic neurosurgery robot in preparation for pre-clinical trials," in *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, IEEE, 2017, pp. 1677–1684.
- [30] K. Gandomi, P. Carvalho, M. Tarasek, E. Fiveland, C. Bhushan, E. Williams, P. Neubauer, Z. Zhao, J. Pilitsis, D. Yeo, *et al.*, "Modeling of interstitial ultrasound ablation for continuous applicator rotation with mr validation," *IEEE Transactions on Biomedical Engineering*, 2020.
- [31] J. MacDonell, N. Patel, S. Rubino, *et al.*, "Magnetic resonance-guided interstitial high-intensity focused ultrasound for brain tumor ablation.," *Neurosurg Focus*, vol. 44, no. 2, E11, Feb. 2018.

- [32] J. MacDonell, N. Patel, G. Fischer, *et al.*, “Robotic Assisted MRI-Guided Interventional Interstitial MR-Guided Focused Ultrasound Ablation in a Swine Model,” *Neurosurgery*, vol. 84, no. 5, pp. 1138–1148, May 2019.
- [33] C. Bove, “Constrained motion planning system for mri-guided, needle-based, robotic interventions,” M.S. thesis, Worcester Polytechnic Institute, 2018.
- [34] S. Pieper, M. Halle, and R. Kikinis, “3d slicer,” in *2004 2nd IEEE international symposium on biomedical imaging: nano to macro (IEEE Cat No. 04EX821)*, IEEE, 2004, pp. 632–635.
- [35] R. Kikinis, S. D. Pieper, and K. G. Vosburgh, “3d slicer: A platform for subject-specific image analysis, visualization, and clinical support,” in *Intra-operative Imaging and Image-Guided Therapy*, F. A. Jolesz, Ed. New York, NY: Springer New York, 2014, pp. 277–289, ISBN: 978-1-4614-7657-3. DOI: 10.1007/978-1-4614-7657-3\_19. [Online]. Available: [https://doi.org/10.1007/978-1-4614-7657-3\\_19](https://doi.org/10.1007/978-1-4614-7657-3_19).
- [36] A. Fedorov, R. Beichel, J. Kalpathy-Cramer, J. Finet, J.-C. Fillion-Robin, S. Pujol, C. Bauer, D. Jennings, F. Fennessy, M. Sonka, *et al.*, “3d slicer as an image computing platform for the quantitative imaging network,” *Magnetic resonance imaging*, vol. 30, no. 9, pp. 1323–1341, 2012.
- [37] T. Farid, K. R. Dhruv, S. Benjamin, Z. Zhanyue, G. Katie, C. S. Shreyas, P. Julie, N. Christopher, and F. Gregory, “Neuroplan: A surgical planning toolkit for an mri-compatible stereotactic neurosurgery robot,” in *2021 International Symposium on Medical Robotics, ISMR 2021*, IEEE, 2021 (submitted).
- [38] J. Widder, “The origins of radiotherapy: Discovery of biological effects of x-rays by freund in 1897, kienböck’s crucial experiments in 1900, and still it is the dose,” *Radiotherapy and Oncology*, vol. 112, no. 1, pp. 150–152, 2014,

ISSN: 0167-8140. DOI: <https://doi.org/10.1016/j.radonc.2014.06.010>.  
[Online]. Available: <https://www.sciencedirect.com/science/article/pii/S0167814014002680>.

- [39] V. P. B. Grover, J. M. Tognarelli, M. M. E. Crossey, I. J. Cox, S. D. Taylor-Robinson, and M. J. W. McPhail, “Magnetic resonance imaging: Principles and techniques: Lessons for clinicians,” eng, *Journal of clinical and experimental hepatology*, vol. 5, no. 3, pp. 246–255, Sep. 2015, S0973-6883(15)00415-6[PII], ISSN: 0973-6883. DOI: 10.1016/j.jceh.2015.08.001. [Online]. Available: <https://doi.org/10.1016/j.jceh.2015.08.001>.
- [40] S. A. Huettel, A. W. Song, G. McCarthy, *et al.*, *Functional magnetic resonance imaging*. Sinauer Associates Sunderland, MA, 2004, vol. 1.
- [41] T. M. Buzug, “Computed tomography,” in *Springer handbook of medical technology*, Springer, 2011, pp. 311–342.
- [42] D. J. Brenner and E. J. Hall, “Computed tomography—an increasing source of radiation exposure,” *New England journal of medicine*, vol. 357, no. 22, pp. 2277–2284, 2007.
- [43] T. L. Szabo, *Diagnostic ultrasound imaging: inside out*. Academic press, 2004.
- [44] D. L. Bailey, M. N. Maisey, D. W. Townsend, and P. E. Valk, *Positron emission tomography*. Springer, 2005, vol. 2.
- [45] D. Chapman, W. Thomlinson, R. Johnston, D. Washburn, E. Pisano, N. Gmür, Z. Zhong, R. Menk, F. Arfelli, and D. Sayers, “Diffraction enhanced x-ray imaging,” *Physics in Medicine & Biology*, vol. 42, no. 11, p. 2015, 1997.
- [46] J. A. Seibert, “X-ray imaging physics for nuclear medicine technologists. part 1: Basic principles of x-ray production,” *Journal of nuclear medicine technology*, vol. 32, no. 3, pp. 139–147, 2004.

- [47] M. Larobina and L. Murino, "Medical image file formats," *Journal of digital imaging*, vol. 27, no. 2, pp. 200–206, 2014.
- [48] *MRI scanner labelled wikipedia article*, Sep. 2020. [Online]. Available: [https://en.wikipedia.org/wiki/File:Mri\\_scanner\\_schematic\\_labelled.svg](https://en.wikipedia.org/wiki/File:Mri_scanner_schematic_labelled.svg).
- [49] S. Currie, N. Hoggard, I. J. Craven, M. Hadjivassiliou, and I. D. Wilkinson, "Understanding MRI: Basic MR physics for physicians," *Postgraduate medical journal*, vol. 89, no. 1050, pp. 209–223, 2013.
- [50] M. Markl and J. Leupold, "Gradient echo imaging," *Journal of Magnetic Resonance Imaging*, vol. 35, no. 6, pp. 1274–1289, 2012.
- [51] N. K. Bangerter, B. A. Hargreaves, G. E. Gold, D. T. Stucker, and D. G. Nishimura, "Fluid-attenuated inversion-recovery SSFP imaging," *Journal of Magnetic Resonance Imaging*, vol. 24, no. 6, pp. 1426–1431, 2006. DOI: <https://doi.org/10.1002/jmri.20743>. eprint: <https://onlinelibrary.wiley.com/doi/pdf/10.1002/jmri.20743>. [Online]. Available: <https://onlinelibrary.wiley.com/doi/abs/10.1002/jmri.20743>.
- [52] *BRAVO - MR application*. [Online]. Available: <https://www.gehealthcare.com/products/magnetic-resonance-imaging/mr-applications/bravo>.
- [53] *CT scan diagram*. [Online]. Available: <https://computertomography.weebly.com/ct-scan-diagram.html>.
- [54] W. A. Kalender, "X-ray computed tomography," *Physics in Medicine & Biology*, vol. 51, no. 13, R29, 2006.
- [55] M. M. Nadrljanski, *Computed tomography: Radiology reference article*. [Online]. Available: <https://radiopaedia.org/articles/computed-tomography?lang=us>.

- [56] U. Themes, *Tomography*, Apr. 2016. [Online]. Available: <https://radiologykey.com/tomography-3/>.
- [57] W. Schroeder, L. Avila, and W. Hoffman, "Visualizing with vtk: A tutorial," *IEEE Computer Graphics and Applications*, vol. 20, no. 5, pp. 20–27, 2000. DOI: 10.1109/38.865875.
- [58] P. Cignoni, M. Callieri, M. Corsini, M. Dellepiane, F. Ganovelli, and G. Ranzuglia, "MeshLab: an Open-Source Mesh Processing Tool," in *Eurographics Italian Chapter Conference*, V. Scarano, R. D. Chiara, and U. Erra, Eds., The Eurographics Association, 2008, ISBN: 978-3-905673-68-5. DOI: 10.2312/LocalChapterEvents/ItalChap/ItalianChapConf2008/129–136.
- [59] R. A. Drebin, L. Carpenter, and P. Hanrahan, "Volume rendering," *SIGGRAPH Comput. Graph.*, vol. 22, no. 4, pp. 65–74, Jun. 1988, ISSN: 0097-8930. DOI: 10.1145/378456.378484. [Online]. Available: <https://doi.org/10.1145/378456.378484>.
- [60] C. Link, H. Pfister, J. Hardenbergh, J. Knittely, H. Lauery, and L. Seilery, *The volumepro real-time ray-casting system*, 2014.
- [61] B. Preim and C. Botha, "Chapter 4 - image analysis for medical visualization," in *Visual Computing for Medicine (Second Edition)*, B. Preim and C. Botha, Eds., Second Edition, Boston: Morgan Kaufmann, 2014, pp. 111–175, ISBN: 978-0-12-415873-3. DOI: <https://doi.org/10.1016/B978-0-12-415873-3.00004-3>. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/B9780124158733000043>.
- [62] A. Lasso, *Watershed, fast marching, and flood filling effects in segment editor*, Apr. 2017. [Online]. Available: <https://discourse.slicer.org/t/>

watershed-fast-marching-and-flood-filling-effects-in-segment-editor/104.

- [63] P. S. Heckbert, "A seed fill algorithm," *Graphics gems*, vol. 275, pp. 721–722, 1990.
- [64] A. Bilqis and R. Widita, "Comparison of segmentation using fast marching and geodesic active contours methods for bone," in *Journal of Physics: Conference Series*, IOP Publishing, vol. 694, 2016, p. 012044.
- [65] *Fast marching method*, Mar. 2021. [Online]. Available: [https://en.wikipedia.org/wiki/Fast\\_marching\\_method](https://en.wikipedia.org/wiki/Fast_marching_method).
- [66] W. Zhang, K. Doi, M. L. Giger, R. M. Nishikawa, and R. A. Schmidt, "An improved shift-invariant artificial neural network for computerized detection of clustered microcalcifications in digital mammograms," *Medical Physics*, vol. 23, no. 4, pp. 595–601, 1996.
- [67] K. Fukushima and S. Miyake, "Neocognitron: A self-organizing neural network model for a mechanism of visual pattern recognition," in *Competition and cooperation in neural nets*, Springer, 1982, pp. 267–285.
- [68] J. Lin, P. Ligomenides, M. Freedman, and S. Mun, "Application of artificial neural networks for reduction of false-positive detections in digital chest radiographs," in *Proceedings of the Annual Symposium on Computer Application in Medical Care*, American Medical Informatics Association, 1993, p. 434.
- [69] A. S. Lundervold and A. Lundervold, "An overview of deep learning in medical imaging focusing on MRI," *Zeitschrift für Medizinische Physik*, vol. 29, no. 2, pp. 102–127, 2019.

- [70] B. Sahiner, A. Pezeshk, L. M. Hadjiiski, X. Wang, K. Drukker, K. H. Cha, R. M. Summers, and M. L. Giger, “Deep learning in medical imaging and radiation therapy,” *Medical physics*, vol. 46, no. 1, e1–e36, 2019.
- [71] J. H. Lee, B. R. Grant, J. H. Chung, I. Reiser, and M. Giger, “Assessment of diagnostic image quality of computed tomography (ct) images of the lung using deep learning,” in *Medical Imaging 2018: Physics of Medical Imaging*, International Society for Optics and Photonics, vol. 10573, 2018, p. 105731M.
- [72] S. J. Esses, X. Lu, T. Zhao, K. Shanbhogue, B. Dane, M. Bruno, and H. Chandarana, “Automated image quality evaluation of t2-weighted liver mri utilizing deep learning architecture,” *Journal of Magnetic Resonance Imaging*, vol. 47, no. 3, pp. 723–728, 2018.
- [73] G. Delso, F. Wiesinger, L. I. Sacolick, S. S. Kaushik, D. D. Shanbhag, M. Hüllner, and P. Veit-Haibach, “Clinical evaluation of zero-echo-time mr imaging for the segmentation of the skull,” *Journal of Nuclear Medicine*, vol. 56, no. 3, pp. 417–422, 2015.
- [74] N. K. Subbanna, D. Rajashekar, B. Cheng, G. Thomalla, J. Fiehler, T. Arbel, and N. D. Forkert, “Stroke lesion segmentation in FLAIR MRI datasets using customized markov random fields,” *Frontiers in neurology*, vol. 10, p. 541, 2019.
- [75] M. Havaei, A. Davy, D. Warde-Farley, A. Biard, A. Courville, Y. Bengio, C. Pal, P.-M. Jodoin, and H. Larochelle, “Brain tumor segmentation with deep neural networks,” *Medical image analysis*, vol. 35, pp. 18–31, 2017.
- [76] *NVIDIA clara*, Apr. 2021. [Online]. Available: <https://developer.nvidia.com/clara>.

- [77] J. Sanders and E. Kandrot, *CUDA by example: an introduction to general-purpose GPU programming*. Addison-Wesley Professional, 2010.
- [78] *AI-assisted annotation*. [Online]. Available: <https://docs.nvidia.com/claras/claras-train-sdk/aiaa/index.html>.
- [79] K. Maninis, S. Caelles, J. Pont-Tuset, and L. Gool, “Deep extreme cut: From extreme points to object segmentation,” *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 616–625, 2018.
- [80] A. Myronenko, “3d mri brain tumor segmentation using autoencoder regularization,” in *International MICCAI Brainlesion Workshop*, Springer, 2018, pp. 311–320.
- [81] *Acoustic medsystems, inc*. [Online]. Available: <https://www.acousticmed.com/>.
- [82] J. MacDonell, N. Patel, S. Rubino, G. Ghoshal, G. Fischer, E. C. Burdette, R. Hwang, and J. G. Pilitsis, “Magnetic resonance-guided interstitial high-intensity focused ultrasound for brain tumor ablation,” *Neurosurgical focus*, vol. 44, no. 2, E11, 2018.
- [83] W. Shang and G. S. Fischer, “A high accuracy multi-image registration method for tracking mri-guided robots,” in *Medical Imaging 2012: Image-Guided Procedures, Robotic Interventions, and Modeling*, International Society for Optics and Photonics, vol. 8316, 2012, p. 83161V.
- [84] K. Gandomi-Bernal, “Towards model based conformal ablation using robotically driven interstitial ultrasound,” Ph.D. dissertation, Worcester Polytechnic Institute, 2021.



- [85] G. Ghoshal, L. Gee, T. Heffter, E. Williams, C. Bromfield, L. Rund, J. M. Ehrhardt, C. J. Diederich, G. S. Fischer, J. G. Pilitsis, *et al.*, “A minimally invasive catheter-based ultrasound technology for therapeutic interventions in brain: Initial preclinical studies,” *Neurosurgical focus*, vol. 44, no. 2, E13, 2018.
- [86] J. MacDonell, N. Patel, S. Rubino, G. Ghoshal, G. Fischer, E. C. Burdette, R. Hwang, and J. G. Pilitsis, “Magnetic resonance-guided interstitial high-intensity focused ultrasound for brain tumor ablation,” *Neurosurgical focus*, vol. 44, no. 2, E11, 2018.
- [87] J. G. Smirniotopoulos, F. M. Murphy, E. J. Rushing, J. H. Rees, and J. W. Schroeder, “Patterns of contrast enhancement in the brain and meninges,” *Radiographics*, vol. 27, no. 2, pp. 525–551, 2007.
- [88] S. N. Gandhi, M. A. Brown, J. G. Wong, D. A. Aguirre, and C. B. Sirlin, “Mr contrast agents for liver imaging: What, when, how,” *Radiographics*, vol. 26, no. 6, pp. 1621–1636, 2006.
- [89] P. Touska and S. E. Connor, “Recent advances in mri of the head and neck, skull base and cranial nerves: New and evolving sequences, analyses and clinical applications,” *The British journal of radiology*, vol. 92, no. 1104, p. 20190513, 2019.
- [90] A. N. Glud, J. Bech, L. Tvilling, H. Zaer, D. Orłowski, L. M. Fitting, D. Ziedler, M. Geneser, R. Sangill, A. K. O. Alstrup, *et al.*, “A fiducial skull marker for precise mri-based stereotaxic surgery in large animal models,” *Journal of neuroscience methods*, vol. 285, pp. 45–48, 2017.
- [91] M. Gerstein and F. Richards, “Delaunay triangulation,” 2006.
- [92] H. Edelsbrunner and E. P. Mücke, “Three-dimensional alpha shapes,” *ACM Transactions on Graphics (TOG)*, vol. 13, no. 1, pp. 43–72, 1994.

- [93] N. Amenta, S. Choi, and R. K. Kolluri, “The power crust,” in *Proceedings of the sixth ACM symposium on Solid modeling and applications*, 2001, pp. 249–266.
- [94] *Brain tumor annotation model*. [Online]. Available: [https://ngc.nvidia.com/catalog/models/nvidia:med:clara\\_mri\\_annotation\\_brain\\_tumors\\_t1ce\\_tc\\_no\\_amp](https://ngc.nvidia.com/catalog/models/nvidia:med:clara_mri_annotation_brain_tumors_t1ce_tc_no_amp).
- [95] B. H. Menze, A. Jakab, S. Bauer, J. Kalpathy-Cramer, K. Farahani, J. Kirby, Y. Burren, N. Porz, J. Slotboom, R. Wiest, *et al.*, “The multimodal brain tumor image segmentation benchmark (brats),” *IEEE transactions on medical imaging*, vol. 34, no. 10, pp. 1993–2024, 2014.
- [96] K. Gandomi-Bernal, “Implementation of a modular software architecture on a real-time operating system for generic control over mri compatible surgical robots,” M.S. thesis, Worcester Polytechnic Institute, 2018.
- [97] F. Tavakkolmoghammad, *Neuroplan: A surgical planning toolkit for an mri-compatible stereotactic neurosurgery robot*, May 2021. [Online]. Available: <https://www.youtube.com/watch?v=m7ytZRURLXo>.