Risk-Based Path Planning for a Steerable Flexible Probe for Neurosurgical Intervention

Tyler Barett
Department of Software Engineering
University of Maryland, College Park
USA
tbarret2@umd.edu

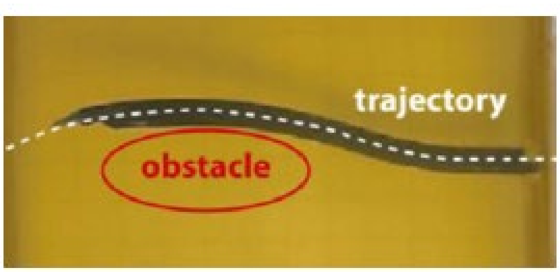
Mahima Arora
Department of Robotics Engineering
University of Maryland, College Park
USA
marora1@umd.edu

***Abstract*— This paper implements an RG-RRT or Reachability Guided Rapidly Exploring Random Tree algorithm for planning the path to reach a goal point inside the brain. It is intended to be used for minimally invasive Neurosurgical Intervention using a steerable flexible probe which has its own kinematic and physical constraints that we are taking care of. Moreover, this algorithm takes care of the weights or costs associated with taking a specific path for e.g., path with higher risks or less clearance from the blood vessels will have a higher cost as compared to the safer path. It also takes the length of the path as the cost when calculating the final path for reaching the tumor. The results are shown on a 2D coronal slice of the brain which has been transformed using scaling, thresholding and contouring depending on our requirements.**

Keywords—RRT, RG-RRT, non-holonomic constraints, computer vision, minimally invasive surgery, tumor

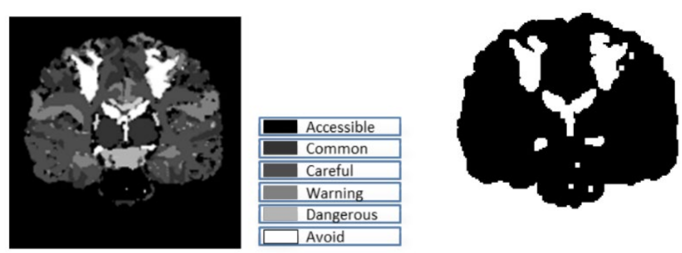
I. Introduction

Our brain is packed with 100 billion neurons compartmentalized into complex bundles of nerves and structures that define who we are. These parts of our brain help us to speak, move, think, learn, and feel emotions. The sheer number and fragility of various components of the brain make navigation during invasive surgery extremely important.
Typical surgery on the brain will use a "keyhole" approach, which minimizes the size of the entry point. This is less harmful to the patient, allowing them to recover faster and experience less pain overall[1]. Through this keyhole, a small manipulator is inserted (usually a steerable flexible needle as shown in the image below) until it makes contact with the mass the surgeon is eliminating. The method is generally preferred but can be difficult to perform. The current industry-standard instruments have a limited range of motion (such as a completely rigid body) or are difficult to manipulate without dedicated planning and control software. This creates the need for software that can



generate a path plan for this manipulator that is safe, smooth, and relatively short.

There is extensive literature that explores the problem of navigating through the brain for neurosurgery using a steerable flexible probe that allows the surgeon to access parts of the brain without damaging tissues and important nerves around the tumor. Research revolves around minimally invasive methods along with getting an optimal path in terms of different costs associated with it like the length of the path taken, clearance from the vessels or important nerves as well as risk involved in moving through a particular region. These weights or costs would be then used by the surgeon to determine which path should be taken to tackle the problem.
In this implementation, we designed a "risk map" based on parameters like taking the shortest path as well as avoiding no-go areas in the brain. We are testing it on an 2D brain map created in OpenCV and the expected results should show multiple paths to the goal but highlight the optimal path which takes into consideration the weights as well.



Moreover, at the end we are describing original research that we implemented that works in the opposite way. It detects the tumor and tries to find the most optimal path to the periphery of the brain. This way, the surgeon will know the exact coordinates from where they should operate for not only a minimally invasive surgery, but also taking the least damaging path for blood vessels as well.

II. Implementation

A. Theory

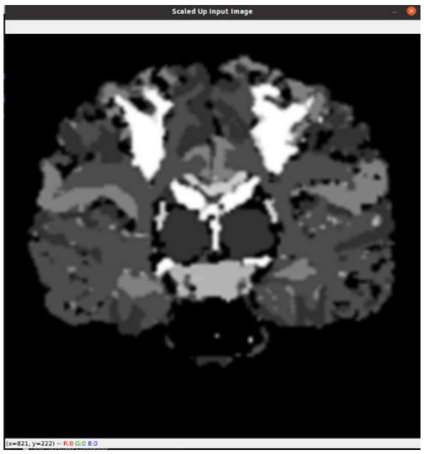
For creating the obstacle map, an image of a coronal slice of brain is taken. As described in Bano's paper[2], we have represented areas of brain based on the risk of damaging the vessels and based on that risk image, we have created a cost map.

Higher the risk, higher will be the cost map. We have used OpenCV library in python to implement this.

B. Creating an Obstacle Map

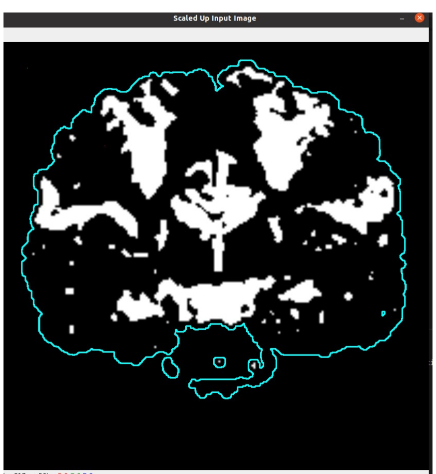
We follow these steps to reach to our desired map on which we are representing the needle's movement using the algorithm

- Take an input image and scale it. In our implementation, we are scaling it 4 times the original image.



- Next, we need to identify the obstacles and associate cost related to these areas. We use OpenCV library for doing thresholding and creating a binary image, white being the "no-go areas" and black being the accessible ones.

- We use contours to define the surface or boundary for the brain.



- Moreover, to identify the points inside and outside the brain, we need to use polyfill functionality in cv2 which gives the following result. Although we are not using this image for plotting the paths as it is featureless , it is useful to get coordinate location(inside or outside the brain).

- The table below represents the cost with respect to each area. The cost will vary from 0 to 1, 1 being the areas with the highest risk. These cost would be useful to calculate the final optimal path.

Area	**Cost**
Accessible	0
Common	0.1
Careful	0.7
Warning	0.8
Dangerous	0.9
Avoid	1
Other Regions	0.6

III. Algorithm

The algorithm used in this paper is an expansion of the common rapidly exploring random trees (RRT) algorithm. This algorithm, however, is adjusted to compensate for the non-holonomic constraints of the probe itself. Hence, this algorithm is named Reachability-Guided RRT (RG-RRT)[3].

At its core, this algorithm behaves much like RRT. At every iteration, a new point in the planning space is chosen. Instead of expanding the closest node by a set maximum distance, RG-RRT will draw a curve with an unbounded length. This directly connects the randomly sampled point and the closest open node on the tree. The curve that connects these two points must be tangential to the point in the tree it is connecting to. Otherwise, it would require a rapid and unrealistic change in pose for the probe. Additionally, this curve must be described with a circle of minimum radius. The probe has an upper bound with which it can rotate. This radius of curvature must be known so that any impossible curves can be rejected.

There is a goal bias which provides a constant 20% chance to randomly choose the goal point rather than a randomly sampled point. This results in periodic checks to see if the tree can connect directly to its goal point.

The reachability tree grows until it is finally able to reach the goal point. This becomes one solution to the problem. This experiment is run multiple times to generate multiple solutions for this specific start and goal position. After multiple solutions are generated, the best path is selected by comparing the cost value of each.

The cost of each path is determined in the Thoreson paper by three factors: the length of the path, the accumulated risk over the path, and the distance of the path from obstacles. Due to time constraints, the calculation that determines the distance of the path from obstacles has been omitted. Instead, the cost function is only dependent on the length and risk of the path. The path length is self-explanatory: it is the sum of every arc length that is on the path to the goal node. The risk is calculated by interfacing with a risk map of the brain, which describes how critical each portion of the brain is. The total risk for each curve is calculated by sampling along the curve and adding the risk level at every sampling point.

A. *Equations*

We use the following equation while calculating our cost function.

$$Cost_i = \alpha \cdot \frac{\lambda_i}{max(\lambda)} - \beta \cdot \frac{\Upsilon_i}{max(\Upsilon)} + \gamma \cdot \frac{\Delta_i}{max(\Delta)}$$

where the λ is the overall length of the path, the Δ is the risk cost and γ represents the clearance from the no-go areas.

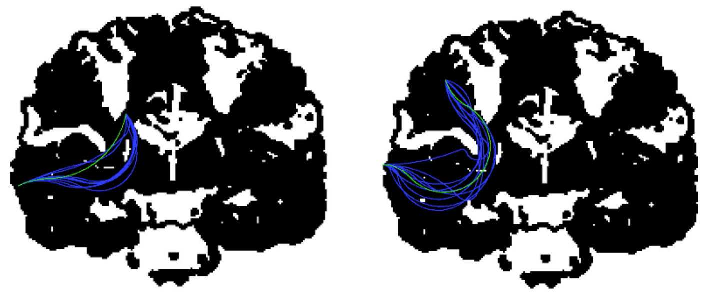
B. *Pseudo Code*

Algoritmo 1 Multiple RRTrees Growth : (qinit, qgoal, MaxTrees)

```
Trees: initialize_MaxTrees(qinit)
 for iter=1:MaxIter
  Qreach=zeros(length(availableTrees));
  while (Qreach(any Tree)=TRUE) do
   prand: Random_Free_State(goal_bias);
   for i=1:availableTrees
        for all q in the Tree(i)
         if Reachable(prand,q,minRadius)
          Qreach(Tree(i))=TRUE
         end if
        end for
   end for
  end while
  for j=1:length(Q_reach)
   qnear(j)=NearestNeighbor(Qreach(j),prand,Tree(Qreach(j)))
   qnew(j)=SolveParameter(qnear(j),prand);
  end for
  [index]=ascending_sort(distance(qnear,prand));
  k=0; ans_ValidEdge=FALSE;
  while (ans_ValidEdge==FALSE && availableTrees>0)
   k=k+1
    ans_ValidEdge=ValidEdge(qnear(index(k)),qnew(index(k)));
  end while
  if ans_ValidEdge
   Tree(index(k)).addNode(qnew(index(k)))
   Tree(index(k)).addEdge(qnew(index(k)),qnear(index(k)))
   if qnew(index(k))==qgoal
        Tree(index(k)).done=TRUE;
   end if
   if all trees reach the goal (i.e. Trees.done==TRUE)
        break
   end if
  end if
 end for
return Trees
```

IV. RESULTS

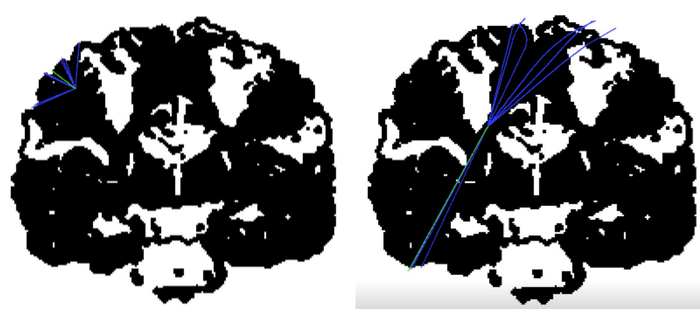
A. *Results from the original paper*



B. *Original Research*

For the original research, our idea was to find a path starting from the tumor and moving outwards to find the best starting point for the surgery. Often, it can be tricky to reach to the goal from one point, but it can be a straightforward path to reach the tumor from another. Hence, our algorithm takes into consideration the same weights as risk areas, length of the path as well as clearance from the "no-go areas" and finds the most optimal path based on the cost returned. We tested our RG-RRT algorithm on six test cases and got an average cost of 23.51 which was 1/10th of the forward algorithm although this might be due to the fact that it tries to find straight and minimally invasive paths while taking other costs into consideration as well.

C. *Results from original Reseach Idea*



V. ACKNOWLEDGEMENT

We would like to express our deep gratitude to Professor Reza Monfaredi and our TAs for the course for their patience, encouragement, and useful feedbacks for this research work.

VI. REFERENCES

[1] Pacific Neuroscience Institute. (2021, April 16). Minimally invasive brain surgery: Keyhole craniotomy. Pacific Brain Tumor Center. Retrieved

May 1, 2022, from https://www.pacificneuroscienceinstitute.org/brain-tumor/treatment/minimally-invasive-brain-surgery/

[2] S. Bano, S. Y. Ko and F. Rodriguez y Baena, "Smooth path planning for a biologically-inspired neurosurgical probe," 2012 Annual International Conference of the IEEE Engineering in Medicine and Biology Society, 2012, pp. 920-923, doi: 10.1109/EMBC.2012.6346082.

[3] Caborni, Chiara, et al. "Risk-based path planning for a steerable flexible probe for neurosurgical intervention." 2012 4th IEEE RAS & EMBS International Conference on Biomedical Robotics and Biomechatronics (BioRob). IEEE, 2012.