Elise Donszelmann-Lund

Bone Cancer Biopsy Navigation

Navigation Math

Given: Tip_{Tool} and $\text{Axis}_{\text{Tool}}$ (tool tip and tool axis in tool frame)
Want: Tip_{CT} and Axis_{CT} (tool tip and tool axis in CT frame)

Requisite Math:

1) Compute Frame Transformation from Tool Frame to Tracker Frame:

   $F_{\text{Track<-Tool}} = T_{\text{Track<-Tool}} * R_{\text{Track<-Tool}}$

   ⇨ Use base vectors from tool frame + center of tracker frame
2) Compute Frame Transformation from Tracker frame to CT frame

   $F_{\text{CT<-Track}} = T_{\text{CT <- Track}} * R_{\text{CT<-Track}}$

   ⇨ Use base vectors from tracker frame + center of CT frame
3) Transform Tool Tip in tracker frame using (1)

   $\text{Tip}_{\text{Track}} = F_{\text{Track<-Tool}} * \text{Tip}_{\text{Tool}}$
4) Compute Tool Axis vector in tracker frame using rotational components of frame transformation

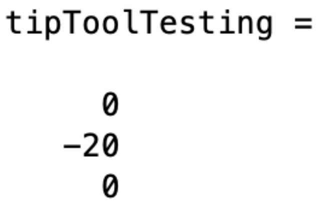
   $\text{Axis}_{\text{Track}} = R_{\text{Track<-Tool}}$
5) Transform Tip and Axis from tracker frame to CT frame

   $\text{Tip}_{\text{CT}} = F_{\text{CT<-Track}} * \text{Tip}_{\text{Track}}$

   $\text{Axis}_{\text{CT}} = F_{\text{CT<-Track}} * \text{Axis}_{\text{Track}}$

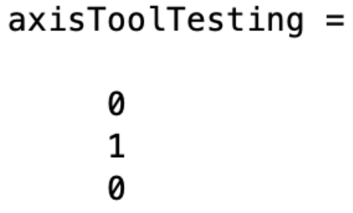
Calibration Testing

Function Output:



```
tipToolTesting =

     0
   -20
     0
```



```
axisToolTesting =

     0
     1
     0
```

Navigation Error Analysis

Discussion of error simulation approaches:

Pure Gaussian distribution with 2s was used on the tracking errors for the 6 markers. This means that the probability of a marker falling outside of the sphere is 5%. 1s would have been 32%, which is far too lenient, and 3s would have been 0.3%, which is far too strict. 5% leaves room for unexpected error (which is bound to occur), but not enough for there to be catastrophic outcomes. Pure Gaussian was used as opposed to Uniform distribution and constant magnitude because a fading marker halo is optimal in this type of problem. We do not want all of the "error" tips or tumor center's to be uniformly in a circle. This is unlikely to happen. Chances are the error in the tool and patient frames will be more centered towards the true location, which is what Gaussian distribution accomplishes.

Overall explanation:

The 3 markers in tool frame and the 3 markers in marker frame (patient skin surface) have error segmented onto them with random gaussian distribution to generate 30 different error poses. The tip of the tool is computed using each of the poses to generate a sphere of tool tip error. The errors are computed as the distance from the "error tips" to the true tip, and the radius of the sphere for tool tip error is sigma of the statistically analysis computation.

The same error segmentation process is applied to the axis vector to create the cone of the tool axis. For each "error axis", the angle is computed. Instead of statistically analyzing for sigma, the maximum error angle is used as the angle of the insertion uncertainty cone. The true axis is rotated using this angle.

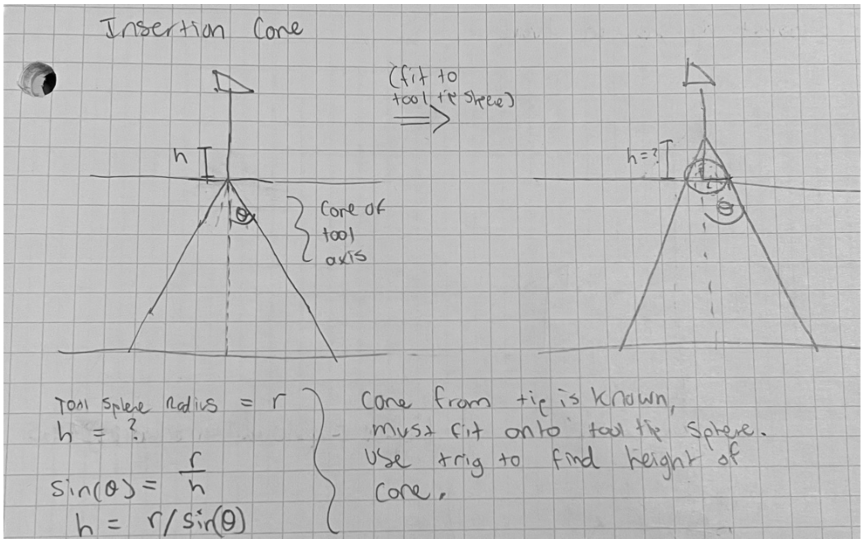
A "cone" can be thought of as a single vector on the outer edge of the cone. This vector was fit along the aforementioned tip sphere using trigonometry to find the starting point of the vector. When this vector intersects with the plane that the tumor center is located on, the distance from the center of the tumor and the intersection is the radius of the tumor. The radius was subtracted by 10% to ensure the cone truly captured all of the tumor.

The sphere of tumor registration error was computed using the error augmented patient marker poses. Each distance from the true tumor center to the cloud of poses was computed and statistically analyzed. Sigma was used as the sphere radius.

The sphere of nerve registration error would have the same radius as the sphere of tumor registration due to the use of the same patient marker poses. The sphere of nerve registration error was allowed to intersect with the sphere of tumor registration error, but could not touch the tumor itself, as this would be in the cone of uncertainty. Therefore, the distance of the closest nerve to hit is the radius of the tumor error cloud STARTING from the outside of the tumor.

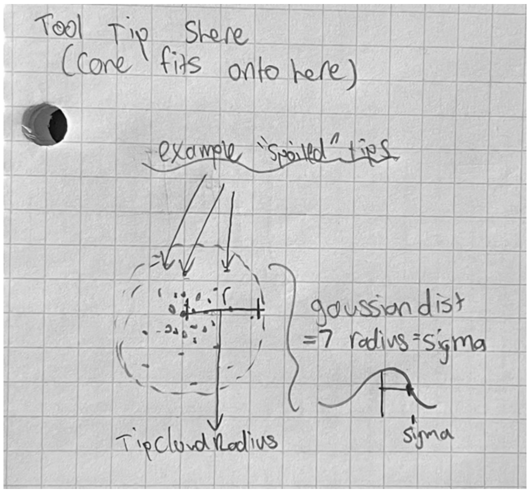
The definition of "near certainty" is defined using the 0.5cm addition between the sphere of tumor registration error and the sphere of nerve registration error. This ensures the sphere's do not actually have a point of intersection and "guarantees" the nerve will not be hit.

Insertion uncertainty cone:



Tool Sphere Radius = r
h = ?
$\sin(\theta) = \frac{r}{h}$
$h = r/\sin(\theta)$

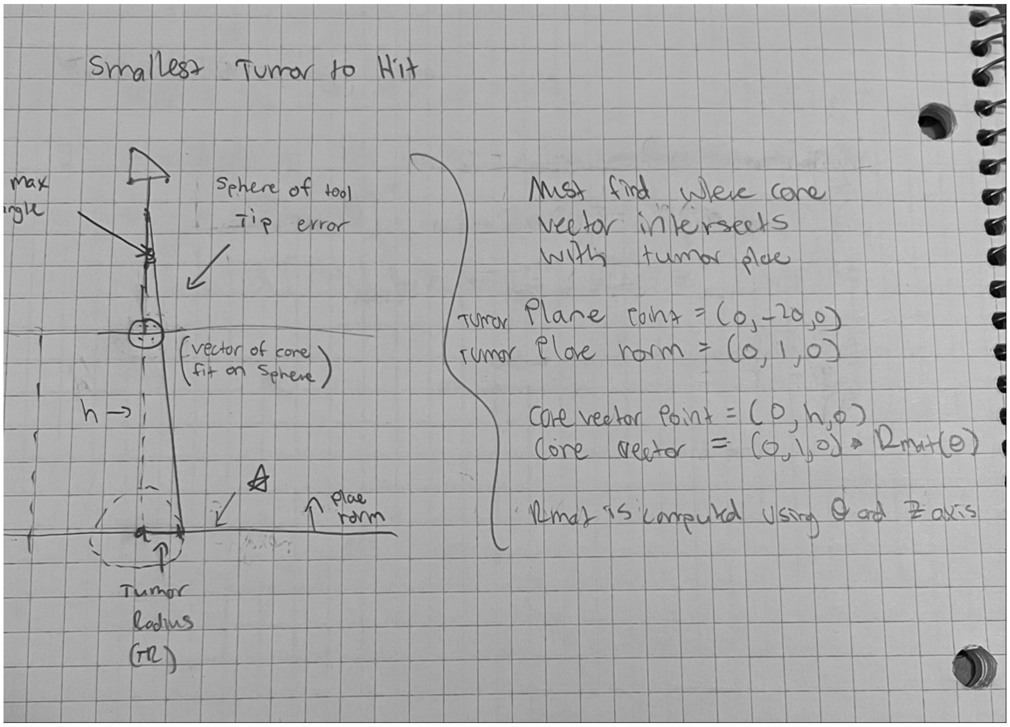
Cone from tip is known, must fit onto tool tip Sphere. Use trig to find height of cone.



Elise Donszelmann-Lund

The angle between each "error axis" and the true axis will be computed for all 30 axes. The max angle is the angle of the cone. The cone's starting point is solved using trigonometry and fit over the tip tool error sphere.

Smallest tumor to hit:



Must find where core vector intersects with tumor plae

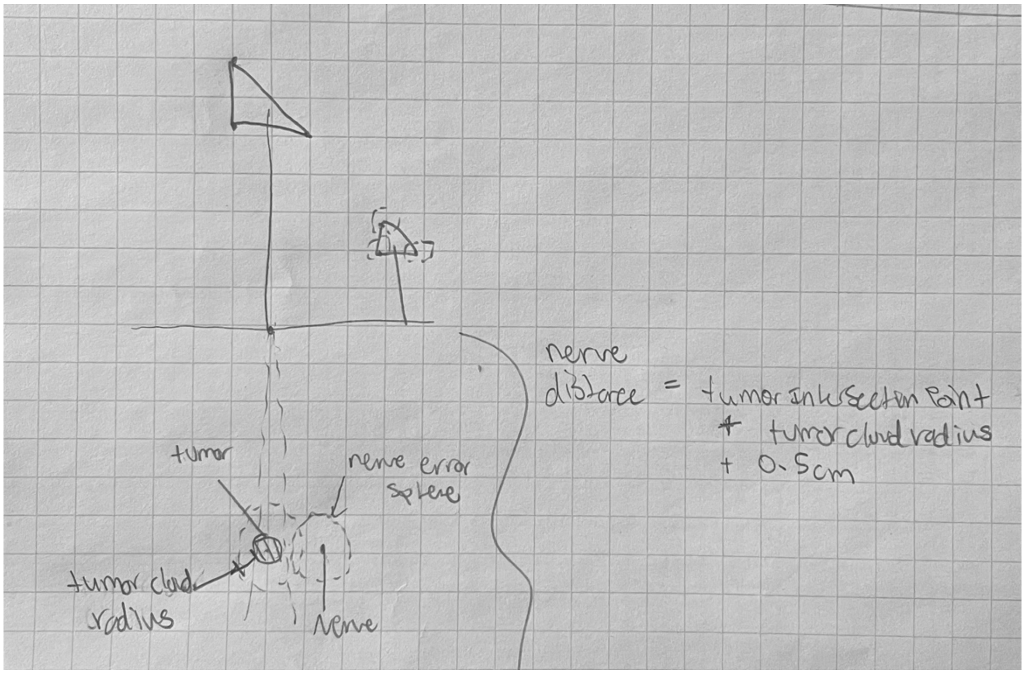
Tumor Plane point = $(0,-20,0)$
tumor Plae norm = $(0,1,0)$

core vector point = $(0,h,0)$
Core vector = $(0,1,0) * Rmat(\theta)$

Rmat is computed using θ and Z axis

The smallest tumor to hit is computed using the point of intersection of the cone to the plane the tumor lies on. The radius is doubled to compute the diameter, and the total diameter is subtracted by 10% to ensure the tumor lies within the cone.

Closest Nerve to Avoid:



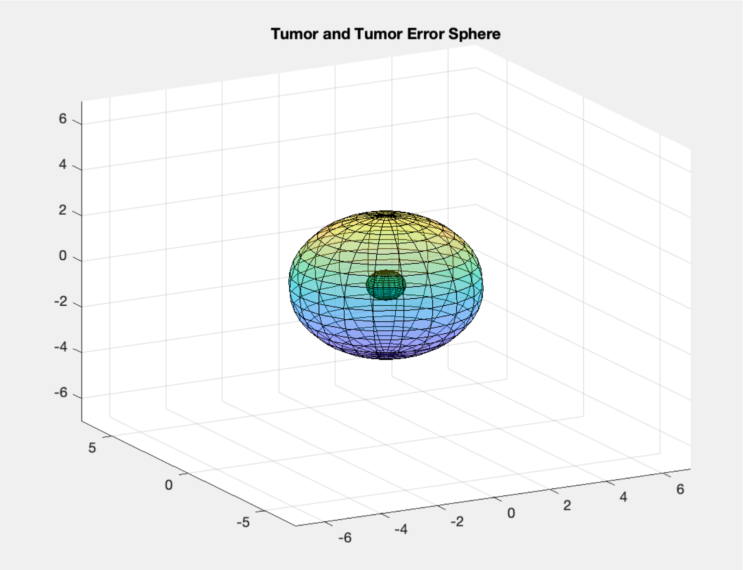
nerve distance = tumor intersection point
+ tumor clud radius
+ 0.5cm

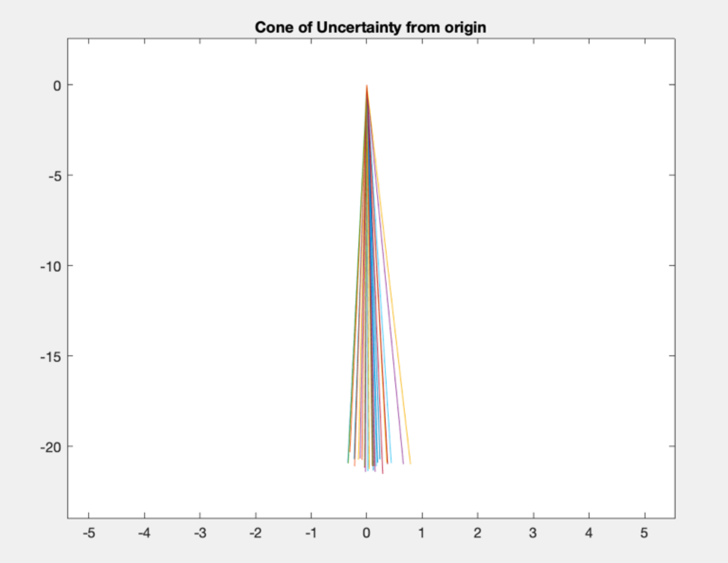
Elise Donszelmann-Lund

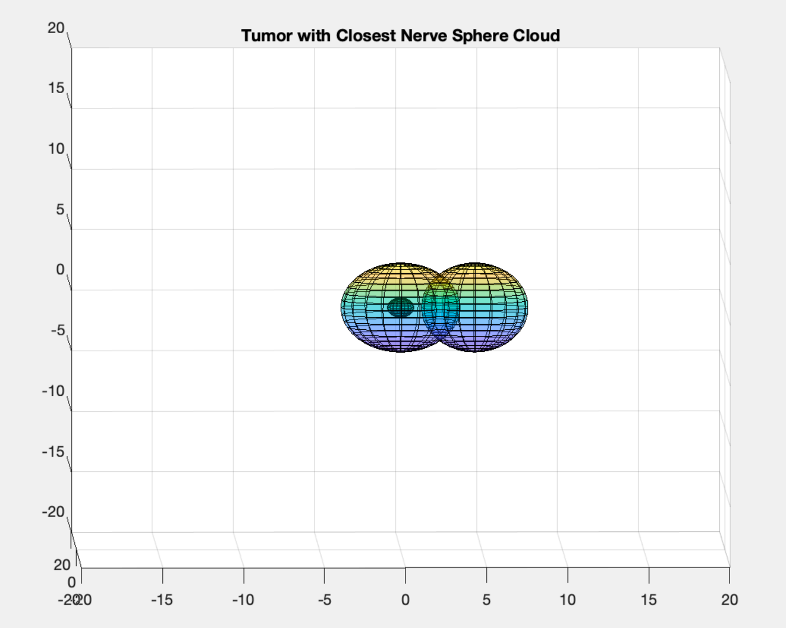
The closest nerve to avoid takes into consideration the tumor's position, the radius of the sphere of nerve registration error, and a "safety" distance addition. The nerve must be outside of the tumor (ie the cone of uncertainty's outer most point), plus 0.5cm to ensure the nerve is safe from being hit. The two registration error spheres are allowed to intersect, but the nerve registration sphere must not intersect with the tumor. This is prevented by adding the radii of the sphere and adding it to the radius of the tumor, and adding the 0.5cm safety distance.

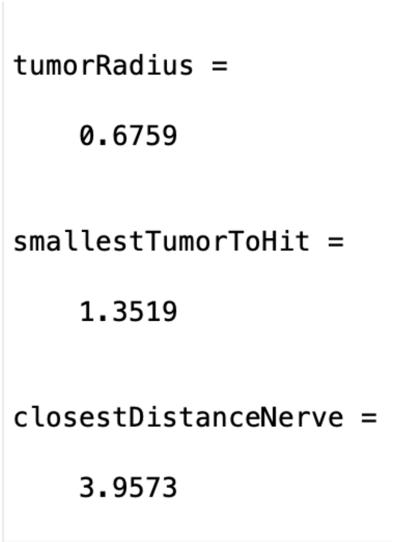
Software Output (example run):

(Figures are centered at the origin for plotting simplicity reasons, these are just meant to demonstrate the general ratios of the tumor, tumor cloud, etc)









```
tumorRadius =

    0.6759


smallestTumorToHit =

    1.3519


closestDistanceNerve =

    3.9573
```

Elise Donszelmann-Lund

Explanation of Results:

The smallest tumor to hit for this trial run was around 1.4cm, and the distance of the closest nerve was around 4cm. This geometrically makes sense. Since the cloud of the nerve registration error is not allowed to come in contact with the cone of uncertainty around the tumor, this ensures the nerve will not be hit during surgery. You can see this visually in the plots. The sphere on the right (sphere of nerve registration error) does not intersect with the small sphere on the left (true tumor). There is also a 0.5cm “safety” zone for extra precaution, which makes the closest distance relatively far compared to the smallest tumor to hit.