

Tomographic Data Analysis

Matthew Ginley-Hidinger, Courtney Hutton, Erika May, James Slaney

May 6, 2015

1 Abstract

This project explores ways in which volume and the surface area of the tumor can be evaluated given theoretical tomographic data. Using both physical and theoretical 3D modeling, this project not only attempts to observe the relationship between the quality of data and the accuracy of the calculated volume and surface area, but also the affect of noise in extracting the area of interest using density to create isosurfaces.

2 Introduction

This paper is intended to address the following question supplied by DEKA, a research and development corporation based in New Hampshire.

”Given theoretical tomographic data for the region surrounding a tumor, can we extract the volume and the surface area of the tumor?”

Our interpretation of the problem is as follows:

Given a volume of interest wherein we know the coordinates of some number of points, can we accurately differentiate between different subvolumes?

To elaborate further, tomographic data consists of a volume with a given number of points, or vertices, that are known to exist at given co-ordinates. Our goal is to use the point density, or the number of points in a given neighborhood around a point, of each point to indicate which subvolume the point is a part of. The ”subvolumes” represent different internal parts of the body. We can use point cloud density to differentiate between the different organs because each organ has a different density value.

Figure 1 shows an example of a computed tomography scan, or CT scan, which use X-Rays for imaging cross sections of different parts of the body.

The structure of the remainder of this paper will be as follows. The second section will describe our initial attempt of solving the problem. We find it appropriate to describe our initial attempt due to the abstract nature of the problem and our unsuccessful attempts

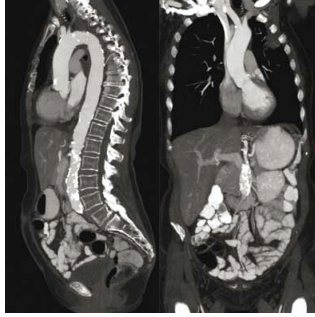


Figure 1: An example of a CT Scan of the abdomen.

at describing our solution best illustrates our modeling process for this problem. In the third section, we will describe how we addressed the problem in actuality. In the fourth, we will present our results. Finally, the fifth section will allow us to conclude the paper. There, we will give ideas for future research, detail the limitations with which we worked, and summarize our methods.

In resolving our problem issued by DEKA, we decided to consider the following questions:

- How do we differentiate between different subvolumes?
- Can identifying points of the same point cloud density of each point lead to an accurate construction of the subvolume?
- How should we use the point cloud densities of the various points? Can we construct an original Python program? Would using isosurfaces to differentiate between the points on a volume of interest from the points on the surface on another volume?
- Can we determine the volume and surface area of the subvolumes of interest once the points on its surface have been identified? Will the measurement be accurate in comparison to known data?
- How does varying the amount of synthesized data affect the accuracy of our measurements?
- Can we distinguish the points that actually are on the surface of a volume of interest as opposed to being a part of another volume or just random faulty data, known as noise?

In our research, we also attempt to create isosurfaces using point cloud density in order to isolate the volume of the tumor. In the interest of our research, an isosurface is defined as a graph of a collection of points in \mathbb{R}^3 that have been assigned a color based on some value. In our case, the value will be point cloud density, where density is specified as the amount of points per volume.

3 Initial Attempts

Since we were not given an explicit data set by DEKA, our main focus was to create our own data by simulating the data point degradation of CT scans. After further correspondence with our industry contact, we were directed to an online repository of data files of real medical scans of the human body. We opened these .obj files using the 3D-imaging functions of Mathematica. Initially, we selected Mathematica as our main software to carry out any calculations or simulations due to our experience with its extensive built-in capabilities. However, after some unsuccessful attempts to implement its surface reconstruction algorithms, we found that these functions were too complex for us to effectively use Mathematica in our research. After exploring other 3D imaging options, we found Meshlab, a 3D mesh processing software. Meshlab has an extensive repository of surface reconstruction and point decimation algorithms which are much more integrated into the software than Mathematica. We tested out these algorithms on the sample point clouds obtained from the online repository. After some initial runs, we again ran into an obstacle: the algorithms on Meshlab were only effective when the decimation and reconstruction algorithms were used in tandem. In other words, we were able to obtain desirable percent errors of volume and surface area if we used the built-in point decimation and surface reconstruction algorithms together. However, when we wanted to create a more realistic simulation by manually eliminating random points, the surface reconstruction created abnormal results due to the loss of the points' normal vectors. Thus, while Meshlab seemed optimal due to its intuitive built-in algorithms and computational abilities, it was not applicable to our project due to its lack of flexibility to produce realistic results.

4 Methodology

After our unsuccessful initial attempts, we revisited the main purpose of this problem. As our industry representative emphasized, we sought to investigate the fidelity of the volume and surface area calculations for a shape as the data set became smaller and smaller. Additionally, we sought to explore the relationship between the density of data points in a region and the ability to discern between two shapes with varying data point densities. Our investigation into this was two-fold: we explored a hypothetical simulation of various data point densities and we completed a physical investigation by generating data from 3D scanned shapes.

4.1 Sphere Isolation

One way we created data was to randomly fill shapes with various amounts of points distributed throughout. In this analysis, we also added points filling a surrounding volume, what we call noise, so that we could determine what was necessary to isolate the shape of interest. In an actual scan, this would be akin to identifying and differentiating different tissues, organs and other objects. This process involved several steps. First, we generated the randomly distributed data. Second, we analyzed the data by calculating the volume

density at each point generated in the previous step. Finally, by analyzing the differences in density, we removed the noise from the sphere.

The volume density was calculated by counting the number of other points within a certain radius from the point of interest. This was done for each point and a histogram was created detailing the number of times each density occurred. This histogram was a Gaussian mixture of two distributions. We hypothesized that one distribution represented the points within the higher density volume of the sphere and the lower density distribution was composed of points that were considered noise. Therefore, if we could separate the two distributions, we could isolate the sphere from the noise.

In order to separate the two distributions, we used a threshold. To decide where the threshold would be set, we fit a Gaussian mixture with two components to the histogram. After translating MatLab’s `gmdistribution.fit` class and method into Python, we used this to fit the double Gaussian to our data. This function uses an expectation-maximization algorithm to modify the parameters iteratively until optimal parameters are found based on least square analysis.

We assume we have the weighted mixture of two probability distributions, shown in Equation 1 and Equation 2. These are Gaussian distributions which rely on two parameters, σ and μ , where σ is standard deviation and μ is the mean. The weighted combination of the two distributions is shown in equation 3, where π represents the respective weights. Here, μ , σ and π are the parameters that are optimized by our expectation-maximization algorithm. Starting with an initial guess for the parameters, the parameters are adjusted until the fit reaches a local maximum, as calculated by least squares.

$$f_1(x|p) = \frac{1}{\sigma_1\sqrt{2\pi}} \exp \frac{-(x - \mu_1)^2}{2\sigma_1^2} \tag{1}$$

$$f_2(x|p) = \frac{1}{\sigma_2\sqrt{2\pi}} \exp \frac{-(x - \mu_2)^2}{2\sigma_2^2} \tag{2}$$

$$f(x|p) = \pi_1 f_1(x|p) + \pi_2 f_2(x|p) \tag{3}$$

Once we fit the Gaussian, we wanted to find the local minimum between the two peaks as the threshold point. To do this, we used calculus and found the zeros of the first derivative. The threshold could then be set at the minimum. If the two peaks were greatly separated, there is a middle part with little change in slope. Because of this, when finding the zeros, we want the zero that is closest to the peak which represents the sphere. Once the threshold was set, we removed the points that had densities below the threshold as our noise was of lower density than the sphere.

4.2 Physical

MeshLab is an open-source 3D mesh processing software that we downloaded from SourceForge. This program features numerous built-in functions and capabilities, including surface reconstruction, data decimation, and computation of geometric measures. Given that our

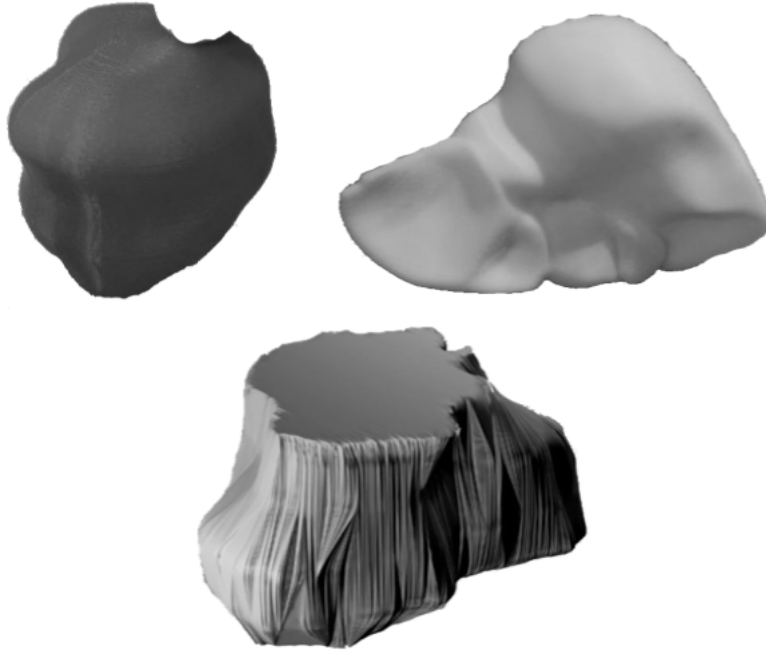


Figure 2: 3D-printed shapes, clockwise from left to right: lung tumor in black, a liver in white, a glioma in black.

final experiment only made use of the latter, we will not go into detail about the formulations of the former; however, the formulas used for reconstruction and decimation can be found in the MeshLab documentation.

We elected to simulate computed tomography using 3D scans of 3D shapes produced by our 3D printer. We obtained 3D files from Thingiverse, an online 3D-mesh repository; selecting a lung tumor, liver, and glioma that had been generated from MRI scans. The shapes were printed in both white and black plastic with a MakerBot 3D Printer; they can be seen in Figure 2. The MakerBot scanner comes equipped with three built-in scan settings for differently colored objects: light, medium, and dark. The "light" setting assumes a scan of an object with a maximally reflective surface, while the "dark" setting assumes an object that is minimally reflective. Based on conversations with technicians in the Occidental Critical Making Studio, we knew that white was the optimal color for scanning an object, while black was the worst. Thus, the white shapes were used as control values. Each shape in each color was scanned on each setting twice, for a total of twelve scans per shape.

The point clouds from the 3D scanner were uploaded into MeshLab, and the software was used to compute the geometric measures for each shape on each setting: volume and surface area. This data and the number of points per scan was exported to Excel and used to analyze percent error from the original volume and surface area obtained from the original 3D modeling file.

For the second portion of our physical experiment, the black and white glioma shapes were placed side by side and scanned. While this configuration was also scanned twice on

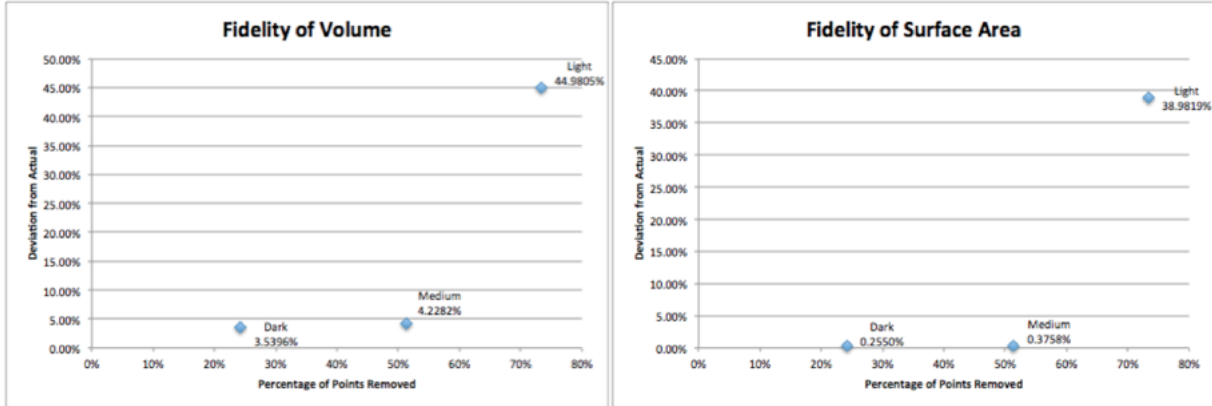


Figure 3: The deviations in volume and in surface area for the black lung tumor, as averaged over multiple trials on three different scan settings.

each setting, only the results from the "light" setting proved usable. The degree of variation seen in the medium and dark scan settings was too large to allow for accurate analysis. The point clouds from these scans were uploaded to CloudCompare, another open-source 3D viewing

5 Results

5.1 Physical

The data from the scans of the black shapes was analyzed in MeshLab to compute the surface area and volume of each shape. The percent error between these values and the known value was then plotted for each shape and scan setting; the graphs can be seen in Figures 3 through 5.

The lung tumor acted as expected; Figure 3 shows that on the light scan setting, the volume deviated from the actual by 44%, while the dark setting only deviated by 3.5%, and the medium setting was approximately the same as the dark setting. A similar trend was observed in the surface area measurements shown in Figure 3. The light setting deviated by 39%, while the dark and medium settings deviated by less than half a percent. Figure 4 depicts the results of the liver trials, which notably did not follow the same trends as the lung tumor. In Figure 4, the volume of the liver was most closely matched by the light setting with only 1.4% error, while the dark and medium setting saw percent errors of 16% and 20% respectively. This was mirrored in the surface area, where the light setting deviated by 1%, while the dark and medium settings varied by 12% and 20% respectively.

The final shape analyzed, the glioma, produced the most surprising results. Figure 5 shows that the volume was again most closely replicated by the light setting, but that the surface area was best matched by the dark setting. Because these deviations are calculated from the average of multiple trials, it is unlikely that this result occurred randomly. The

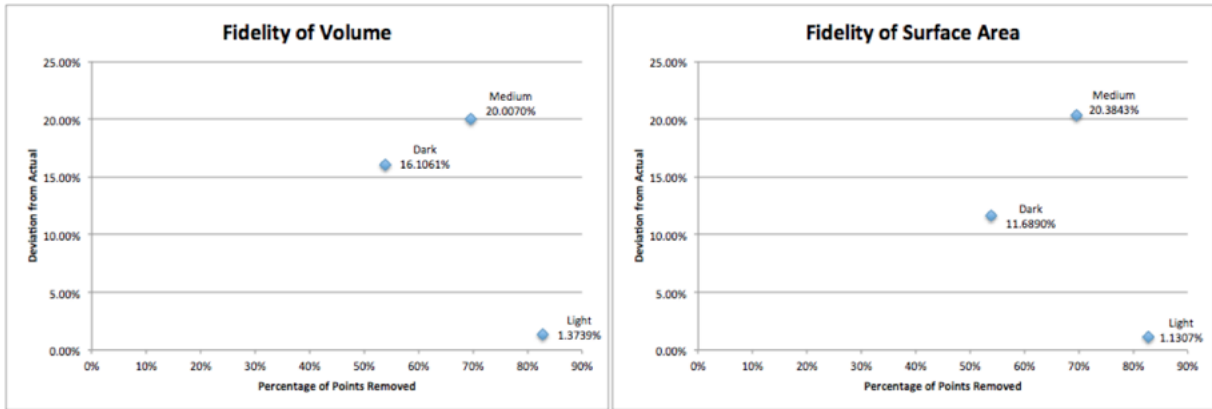


Figure 4: The deviations in volume and in surface area for the black liver tumor, as averaged over multiple trials on three different scan settings.

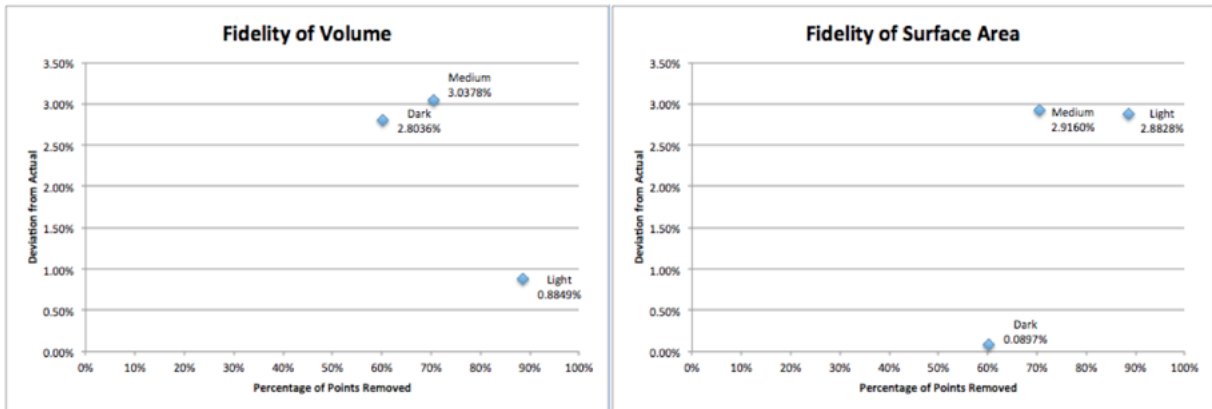


Figure 5: The deviations in volume and in surface area for the black glioma, as averaged over multiple trials on three different scan settings.

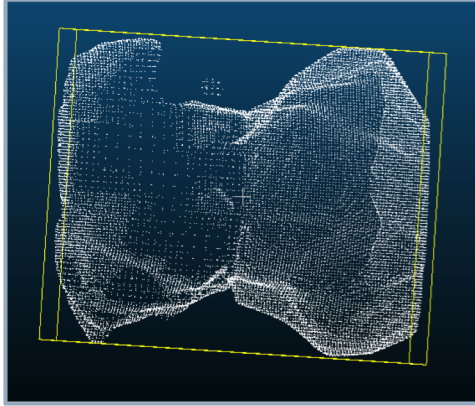


Figure 6: Point cloud generated by a 3D scan of the black and white gliomas.

glioma, shown in Figure 2, had the most irregular surface out of all of our shapes. The majority of the shape was covered with striations, which may explain why the scanner returned different readings for the volume and the surface area. Additionally, it should be noted that the maximum deviation from the actual volume was 3.0%, observed on the medium setting; the maximum deviation from the actual surface area was 2.9%, also observed on the medium setting. While the results from the glioma scans are not consistent with our predictions, the magnitude of error is much smaller than was seen in other trials. Thus, the inconsistencies may be less relevant because they occur on such a small scale. Like the glioma, the liver shown in Figure 2 is also an irregular shape. One side was convex and smooth, the other was concave and irregular, with multiple ridges and divots. As before, this irregularity may account for the unexpected trends in the data, where the light setting returned the most accurate results. Furthermore, the lung tumor in Figure 2 is predominately smooth, without any abrupt edges or indentations; this would account for why this shape returned our expected results.

For the final investigation with our 3D shapes, the black and the white glioma were juxtaposed on the 3D scanner to simulate two objects with varying data point densities. The result of the scan can be seen in Figure 6.

This point cloud was uploaded into CloudCompare, and an isosurface of the data densities was generated by the software, shown in Figure 7. From this surface, there is a clear distinction between the two shapes. On the right, the figure shows an object with a uniform density, which correlates to the largest peak on the histogram. On the left side of the isosurface, the shape clearly has a more irregular density of data points. While there are areas of higher density, these correspond to edges on the glioma, and are to be expected. Given the task of separating two objects with varying data densities, we conclude that it is possible to distinguish the two using only isosurfaces.

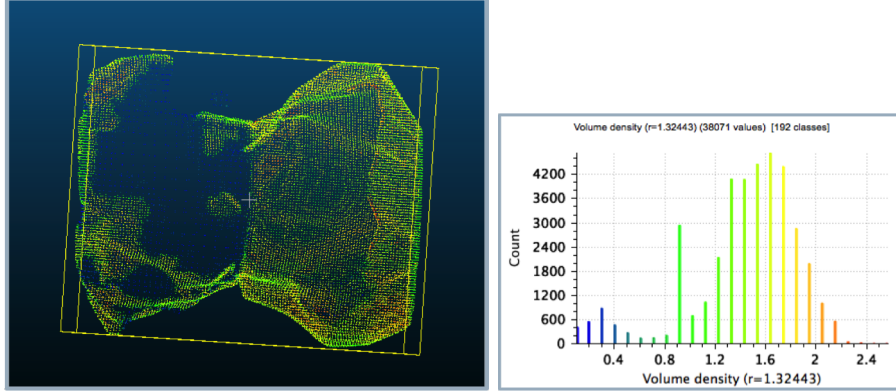


Figure 7: The isosurface and histogram generated by CloudCompare.

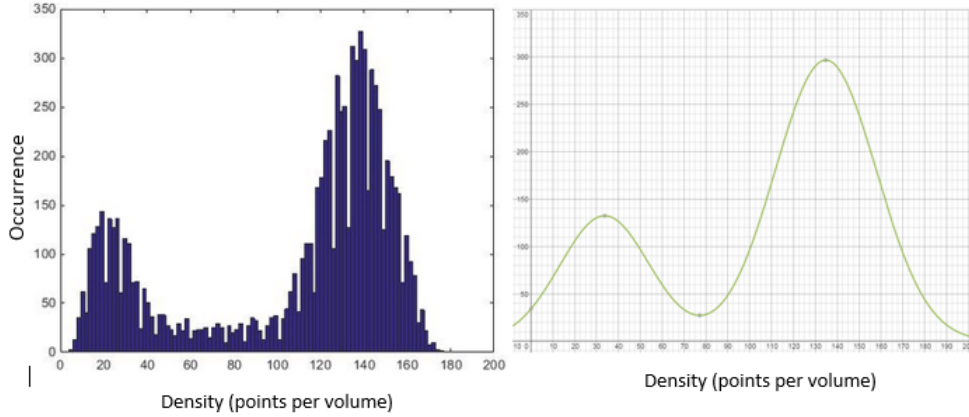


Figure 8: Histogram of the volume density distribution (left) and the corresponding double Gaussian fit (right). The equation for this particular fit can be seen in equation 4. The threshold is set at the minimum of the Gaussian mixture curve.

5.2 Sphere Isolation

By calculating the volume density for each point in a given set of vertices, we created a histogram showing the occurrence of each density and observed two distinct distributions (Figure 8, left). In the particular case depicted in Figure 8, there was a lower peak with a mean of 33.7 and a higher peak with a mean of 134.8. The Gaussian mixture fit algorithm was then able to calculate the respective weight and standard deviation for each peak.

The calculated equation for this distribution is shown in Figure 8. The weight, π , is the numerator of each of the two terms. The standard deviation and mean correspond to their positions in Equations 1 and 2.

$$f(x|p) = \frac{17204}{23.15\sqrt{2\pi}} \exp \frac{-(x - 134.7783)^2}{2(23.15)^2} + \frac{6789}{20.46\sqrt{2\pi}} \exp \frac{-(x - 33.7)^2}{2(20.46)^2} \quad (4)$$

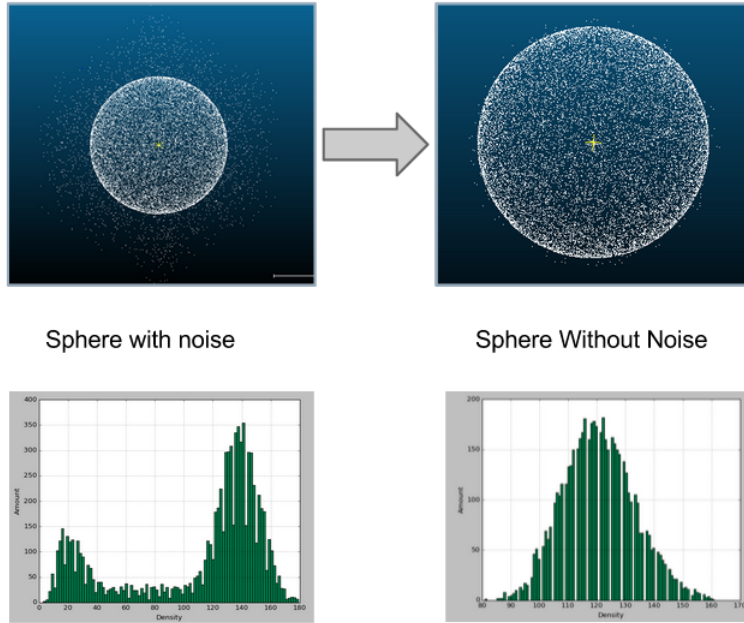


Figure 9: Reduction in noise seen from setting a threshold at the Gaussian minimum. (top) Distribution of vertices seen before (left) and after (right) removing points below the threshold. (bottom) Corresponding density distributions derived from images above.

We can then take the derivative of Equation 4 and calculate the zeros in order to find the local minimum. In this case, the local minimum was 77.16. The threshold was then set at this point and all points which had volume densities lower than this were removed from the set of vertices.

As a result, we saw a lot of reduction in noise. As can be seen in figure 9, most of the outlying points which were not part of the sphere were removed with only a few points close to the surface of the sphere remaining. Also shown in figure 9 is the corresponding density distributions. We see that with a threshold the sphere goes from having the two peak Gaussian mixture to a single Gaussian curve representing the sphere. This indicates that our initial hypothesis was right and that the sphere with randomly distributed points made up one distribution within the mixture.

We also tested this method on a variety of shapes including a torus, a cube and an egg shape. We saw very good results for all these shapes. However, the method does not work as well for shapes which have large variance in their density as this distribution would not be normal and would not be seen as a single peak in the histogram. In this case, it would be more informative to find local clusters within the scope of the shape so that the overall shape could be found based on the smaller components. This is an avenue for further development.

6 Conclusion

6.1 Limitations & Further Research

Without access to actual tomographic data, we elected to generate our own from 3D printed shapes, a 3D scanner and an original Python program. However, unlike a CT scan, a 3D light scan only returns points on the surface of the object. In the future, performing similar experiments with real-world computed tomographic data could support the validity of our conclusions. Narrowing the definition of fidelity or the scope of degradation would also undoubtedly lead to greater precision and accuracy during experimentation. Furthermore, the data from a CT scan includes points on the interior of an object, which provides a much more complex definition of "data point density" than we examined in our experiment. We believe that our model shows that it is possible to create an isosurface from a point cloud, however, our model currently cannot be applied to point clouds produced through computed tomography.

Additionally, the 3D scanner used in our investigation does not emit X-Ray waves, but rather emits light waves. The most significant limitation of this is that it prevent us from investigating the effect of wave probability models. In the future, simplified wave equations such as $P(x) = A \exp(-\alpha x)$ could be applied to tomographic data to include the probability of a particle appearing in a certain place at that point in time.

6.2 Summary

In conclusion, we summarize what we have accomplished. We simulated a set of tomographic data and distinguished between a subvolume of interest and surrounding data, all in an original Python script. Doing so required us to consider the point cloud density for the various points in order for us to construct a double Gaussian curve. We then created a threshold to distinguish which points are most likely on the surface of the sphere. This method was fairly successful considering the graphs in section 5.2.

We also assessed the relationship of how our algorithms for assessing the volume and surface area of volumes of interest corresponded with the amount of data. To do this we compared the computed volume of the file once it was downloaded to our computer to the data that resulted from the scan of the 3D print. In this process, we found results of mixed accuracy. Shapes with the less uniform surfaces such as the liver and the glioma yielded data that was inconsistent with the amount of data that was removed in each scan.

Finally, we worked with isosurfaces to distinguish between two volumes. We scanned a white and a black 3d printout of a glioma together. Black objects do not scan as well in the printer, so less points were gathered for the black object. This rendered the point cloud density of the points on the surface to be low on the black object that they were on the white object. This experiment was mildly successful as there were still some points on the surface of the black glioma that had point cloud densities similar to those on the surface of the white glioma.

7 Appendix

Python code for constructing a sphere:

```
import matplotlib.pyplot as plt
from mpl_toolkits.mplot3d import Axes3D
import math
import random
import numpy as np
import sys
sys.path.append('C:\\Users\\Matt\\Desktop\\')
def produceSphere():
    dataFile = open('C:\\Users\\Matt\\Desktop\\sphere.obj', 'w')
    global rng #rng goes from -rng to +rng
    rng=5
    number=1000
    # ADDING NOISE
    noisePercent = .6
    np=noisePercent
    n=int(number+(number*np))
    x=[0]*n
    y=[0]*n
    z=[0]*n
    for p in range(0,n):

        el=math.asin(2*random.random()-1)
        az=2*math.pi*random.random()
        r=(rng-1)*math.pow(random.random(),1/3)
        x[p]=r*math.cos(el)*math.cos(az)
        y[p]=r*math.cos(el)*math.sin(az)
        z[p]=r*math.sin(el)
        dataFile.write('v' + ' ' + str(x[p]))+' ' + str(y[p])+ ' ' + str(z[p]))
        dataFile.write('\n')
    for noise in range(number,n):
        x[noise]=(2*rng)*random.random()-rng
        y[noise]=(2*rng)*random.random()-rng
        z[noise]=(2*rng)*random.random()-rng
        dataFile.write('v' + ' ' + str(x[noise]))+' ' + str(y[noise])+ ' ' + str(z[noise])
        dataFile.write('\n')
    # plotPoints(x,y,z)
    return x,y,z

def plotPoints(x,y,z):
```

```

fig = plt.figure()
ax = fig.add_subplot(111, projection='3d')
ax.scatter(x,y,z,depthshade=False)
plt.grid()
plt.show()

def densityGrid(x,y,z):
    s=len(x)
    b=20#number of boxes will be b^3
    step=(rng*2)/b
    grid = np.zeros([b,b,b])
    total=0
    for p in range(len(x)):
        f=math.floor((x[p]+rng)*(b/(rng*2)))
        g=math.floor((y[p]+rng)*(b/(rng*2)))
        h=math.floor((z[p]+rng)*(b/(rng*2)))
        grid[f,g,h]+=1
        # print str(f) + ' ' + str(g) + ' ' + str(h)
    return grid

def histo(g):

    # the histogram of the data
    n, bins, patches = plt.hist(g)

    # add a 'best fit' line
    # y = mlab.normpdf( bins, mu, sigma)
    # l = plt.plot(bins, y, 'r--', linewidth=1)

    plt.xlabel('Density')
    plt.ylabel('Amount')
    plt.title(r'')
    plt.grid(True)

    plt.show()

if __name__ == '__main__':

    x,y,z= produceSphere()
    # for point in rng(len(x)):
    plotPoints(x,y,z)
    gr=densityGrid(x,y,z)

```

```

g=gr.flatten()
# for gg in range(len(g)):
#     if gg==len(g)-1:
#         break
#     if g[gg]<5:
#         g=np.delete(g,[gg])
#
# print g[0:20]
# print len(g)
# # print sum(gr)
histo(g)'

```

8 References

CloudCompare. Computer software. CloudCompare. Vers. 2.6.1. EDF R&D, 15 Feb. 2015. Web. 12 Mar. 2015.

Ducros, A., M. Boukobza, R. Porcher, M. Sarov, D. Valade, and M.G. Bousser. "The Clinical and Radiological Spectrum of Reversible Cerebral Vasoconstriction Syndrome. A Prospective Series of 67 Patients." *Brain* (2007): 3091-101. Web. 4 Apr. 2015. JPG.

Edelen, Mark. "Tumor with MRI Slices." Thingiverse.com. MakerBot, 8 Jan. 2015. Web. 4 Apr. 2015. OBJ.

Kane, Derek. "Tomography Clarification." 4 Feb. 2015. E-mail.

Larsen, Richard J., and Morris L. Marx. *Introduction to Mathematical Statistics and Its Applications*. 5th ed. Boston: Prentice Hall, 2012. Print.

Lichty, Patrick. "Salvatore Iaconesi's Glioma." Thingiverse.com. MakerBot, 22 Sept. 2012. Web. 4 Apr. 2015. OBJ.

MeshLab. Computer software. MeshLab Vers. 1.3.3. Visual Computing Lab, ISTI—CNR, 2 Apr. 2014. Web. 1 March 2015.

Mumm, Carina. "Body Parts: Liver." Thingiverse.com. MakerBot, 13 Feb. 2013. Web. 4 Apr. 2015.