Boundary Point Detection from Noisy Tomographic Data

Frank Madrid Jeff Cadena Sebastian Erquiaga Spencer Pruett

PIC Math Project

Department of Mathematics
California State University, Bakersfield

Abstract

A common method to analyze the internal structures of a body is through the use of computed tomographic (CT) imaging in conjunction with computing algorithms. In this project, we consider the problem of detecting surface points of a volume from 3D tomographic data produced by a sampling beam that may return points inside the volume. To detect surface points, we treat points off the surface as "noise" and propose two methods to reduce the presence of noise in the data. We demonstrate our methods on several data sets with varying noise models.

I. Introduction

A fundamental problem in medical imaging is to properly reconstruct the surface of a volume from a given set of randomly sampled points that are located on and possibly within the volume. An accurate reconstruction of the surface of a volume within the human body, such as a malignant tumor, can be used to properly diagnose and treat a patient.

The specific surface reconstruction problem that we consider was proposed by Dr. Derek Kane from DEKA Research and Development. Dr. Kane posed the following question: *Given tomographic data in the region surrounding a tumor, can we extract the volume occupied by the tumor? Specifically, given a randomly selected set of points from a volume, how can we determine the surface enclosing the volume assuming the points are equally likely to occur anywhere within the volume? Furthermore, do our proposed methodologies of determining the surface change when we are more likely to detect elements of the volume which are nearer to the surface?*

In this report, we propose two methods, the *Reverse k-Nearest Neighbors* algorithm [4] and *Robust Cocone* [2], to give a solution to the surface reconstruction problem posed by Dr. Kane. The viewpoint that we take is to treat the points that are sampled near, but not on, the volume as "noise" while the points on the surface as the "clean" data. The two methods we propose reduce the effect of noise while preserving, as much as possible, the surface or boundary points of the volume. We test the methods on data sets obtained from the Japanese Database Center for Life Sciences, as suggest by Dr. Kane.

This report is organized as follows. In section II we provide a formalized description of the problem and the mathematical setup of our methods and the associated mathematical tools. In section III we describe in detail the discrete tomographic models used to solve our problem. In section IV we describe and discuss our methods, data specifications, and computation times. Lastly, in section V, we discuss our results, the limitations of our proposed solution, and possible improvements.

II. Background and Problem Statement

In computer science and applied mathematics, volume surface reconstruction from 3D point clouds is an active area of research [1, 2, 4]. Surface reconstruction techniques produce a surface mesh or triangulation from a finite sample of the surface. A popular algorithm known as the Power Crust algorithm [1] uses an interpolation of the data to generate a surface mesh which contains no holes, shown to be independent of the quality of the input point sample, but is slow when compared to other algorithms. The original Power Crust algorithm, and most existing methods, assumes that the sample points are all on the boundary of the volume, in other words, perfect surface sampling is assumed. Dey improves on the Power Crust algorithm in [2] with the Robust Cocone algorithm which shows a marked improvement in computational time and the ability to accommodate large quantities of noise under reasonable noise models. Lastly, in [4], Xia et. al. implement an efficient utilization of the reverse k-nearest neighbors algorithm, an extension of the k-nearest neighbors algorithm, which they call *BORDER* and show its effectiveness in determining the boundary points of a set of data points which are located at the margin of densely distributed data.

In this report, we consider the surface reconstruction problem in the presence of noisy point samples. To be precise, let V be a volume in 3D space with boundary surface ∂V. Given a finite sample S of V, we are interesting in finding a subset $\tilde{S}$ of S that maximizes the number of points in the set $\tilde{S} \cap \partial V$. In other words, we seek to remove as many points of S that are not on the boundary of V and simultaneously keep as many points in S that are on the boundary ∂V. To solve this problem, we take the viewpoint that points in S that are not on the boundary ∂V are noisy point samples. We propose a modified version of the original BORDER algorithm in [4] to find $\tilde{S}$, and also analyze whether implementing Robust Cocone is advantageous to further detect surface points of the volume. In the following sections, we describe the main algorithms that we used to solve our research problem.

A. k-Nearest Neighbor

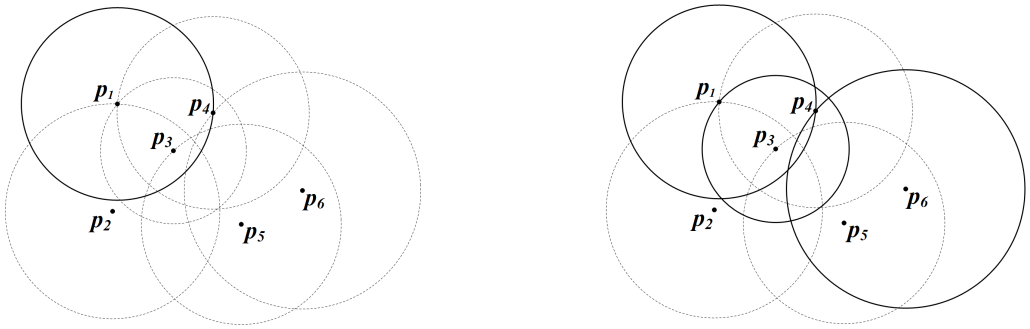
The k-Nearest Neighbor algorithm (kNN) is a non-parametric algorithm used for classification purposes [4]. If k is a natural number then $\text{kNN}_k(p)$ is the set of k-nearest points, measured by the least Euclidean distance, within the sample S to the query point p. Fig. 1a gives an example of the kNN algorithm for $k = 2$ and query point p_1, for a data set $S = \{p_1, p_2, \ldots, p_6\}$. From Fig. 1a, $\text{kNN}_2(p_1) = \{p_3, p_4\}$.

B. Reverse k-Nearest Neighbors

An extension of kNN, the *Reverse k-Nearest Neighbors* algorithm (RkNN) assigns a RkNN value to each point where the RkNN value of a point p is the number of times it appears in the k-nearest neighbor list of all the points in the sample set S. The reverse k-nearest neighbor list of p in S is the set $\text{RkNN}_p(k)$ consisting of points p_i in S such that p is a k-nearest neighbor of p_i. The RkNN value of p is the number of points in $\text{RkNN}_p(k)$. Fig. 1b gives an example of the RkNN algorithm for $k = 2$ and query point p_4. From Fig. 1b, we have that $\text{RkNN}_{p_4}(2) = \{p_1, p_3, p_6\}$ and thus p_4 has a RkNN value of $|\text{RkNN}_{p_4}(2)| = 3$.

A key feature of the reverse k-nearest neighbor algorithm is that it characterizes the neighborhood of a point with respect to the entire sample instead of the point itself.

In this way, the algorithm can capture the distribution property of the underlying data and facilitate the identification of boundary points.



(a) *Example of kNN for point* p_1 *and* $k = 2$.

(b) *Example of RkNN for point* p_4 *and* $k = 2$.

Figure 1

C. Power Crust

Introduced in [1], the Power Crust algorithm considers a finite subset of points from the surface of a three-dimensional volume and produces a surface mesh from an approximation of what is known as the *medial axis* of the volume [**?**]. The algorithm first computes the *medial axis transformation* (MAT) from the set of sample points and then applies an inverse transformation to the medial axis transform to produce a piecewise-linear surface approximation of the surface. The MAT is approximated by a subset of the *Voronoi vertices* of the sample called *poles* which lie near the medial axis. The balls surrounding the poles and touching the nearest sample points are the *polar balls*. The polar balls approximate maximal balls contained in the interor or exterior of the sample space and the radii of the polar balls define weights on the poles. The inverse transform is approximated using the *power diagram* of the set of weighted poles which divides the space into polyhedral cells. A subset of the power diagram cells are represented as the interior of the object. Lastly, the subset of the two-dimensional polygonal faces of the power diagram which separate these inner cells from the outer cells forms the Power Crust output surface.

D. Robust Cocone

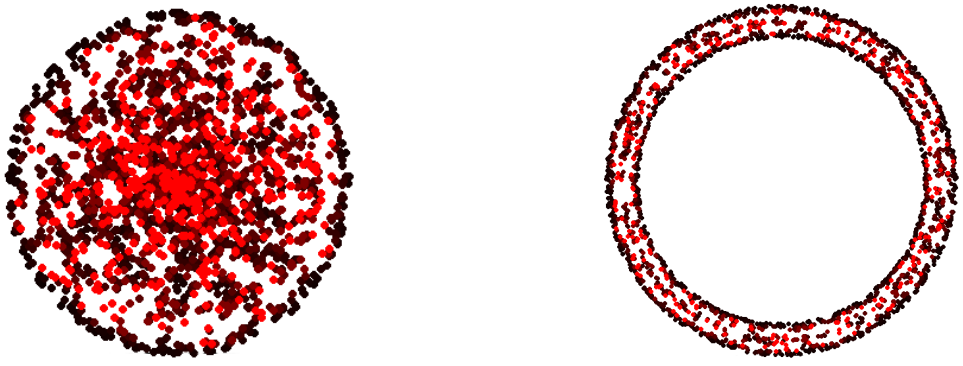
The Robust Cocone algorithm was pioneered by Dey and Goswami [2]. One notable feature of this algorithm is that it is proven to provide an accurate estimation of the boundary of a volume even when the sample space consists of noisy data. The Robust Cocone algorithm uses the *Voronoi diagram* of the sample points and uses methods similar to the Power Crust algorithm. As in the Power Crust algorithm, a subset of the polar balls are classified as large and are categorized as inner and outer polar balls, depending on whether they are inside or outside the volume, respectively. The union of these larger interior and exterior polar balls and the surface are topologically equivalent. As such, the sample points that lie on the large outer balls are retained and the surface is identified from them. The rest of the points in the sample space are discarded.

III. Methodology

In this project, we considered a 3D point cloud of a region surrounding an object where experimental noise was simulated by randomly perturbing a random subset of the sample data simulating noisy data. Our goal was to mitigate the effect of the noise perturbation on the tomographic data. To accomplish this, we implemented the *RkNN* and Robust Cocone algorithms which attempt to remove the noisy points by either exploiting the disparity in density between the clean data and noisy data, or by considering the surface interpolating a subset of the noisy tomographic data. The success of these experiments was quantified by the percentage of noisy points remaining after the experiment while maintaining a minimum percentage threshold of the clean sample points.

A. Method 1: Reverse k-Nearest Neighbors

The authors in [4] proposed a boundary point detector called BORDER which utilizes the special property of the reverse k-nearest neighbors to find the boundary points in a data set. The key idea of the BORDER algorithm is that points on the boundary of a uniformly distributed sample will have a low RkNN value compared to points that lie in the interior of the volume. However, since one of our underlying assumptions is that the strength of the sampling beam decays as it passes through the volume, most of our sample points will exist along the surface of the volume, and therefore we considered points with relatively high RKNN values as candidate boundary points. This difference can be seen in Fig. 2 where we consider two data sets. The data set in Fig. 2a considers points uniformly distributed throughout the sphere where its boundary points have relatively low RkNN values. The data set in Fig. 2b considers points distributed about the surface of the sphere where its boundary points have relatively high RkNN values. Below, we describe the algorithm for our proposed method.



(a) *Uniformly distributed data set within volume* **(b)** *Uniformly distributed data set near surface*

Figure 2: *Variation in RkNN value for uniformly distributed data within and on the surface of a volume*

Let $\mathbf{P}$ be a $m \times 3$ matrix representing the set of m, 3-dimensional tomographic data with noisy points. We first run the kNN algorithm on $\mathbf{P}$ to produce an $m \times k$ matrix $\mathbf{K}$ where $\mathbf{K}_{(i,j)}$ is the index of the j-th closest point to the point with index i and $1 \leq j \leq k$. We then let $\mathbf{B}$ be the set of points having a RkNN value of at least some threshold α. The points $\mathbf{B}$ will serve as our candidate surface points from the original noisy sample $\mathbf{P}$.

Procedure 1 rkNN$(\mathbf{P}, k, \alpha)$

Input: A $m \times n$ matrix $\mathbf{P}, k \in \mathbb{N}, \alpha \in \{1, \ldots, k\}$
Output: A matrix $\mathbf{B}$ of points representing the border points of $\mathbf{P}$

```
K ← knnSearch(P, k)
for i ← 1 to m do
    ReverseCount ← 0
    for j ← 1 to m do
        if i ∈ K_j then                    ▷ where K_j is the row vector j of matrix K
            ReverseCount++
        end if
    end for
    if ReverseCount ≥ α then
        B.add(P_i)                         ▷ where P_i is the point with index i of matrix P
    end if
end for
return B
```

Procedure 1 describes our RkNN approach over a specified range of k and α. We hypothesize that the true surface points will have a significantly higher RkNN value then the noisy points due to the density of the clean points.

In order to improve on the results obtain from our RkNN implementation, we apply the Robust Cocone algorithm on the resulting set of points $\mathbf{B}$ from Procedure 1.

B. Method 2: RkNN + Robust Cocone

The authors in [2] proposed an algorithm to reconstruct a surface of an object represented by a set of points scattered around the object's sampled surface in the presence of noise called Robust Cocone. We implemented Robust Cocone by executing the function *robust cocone-win.exe* provided to us by Dr. Tamal Dey, one of the creators of Robust Cocone, on the set of points $\mathbf{B}$ generated by Procedure 1. The Robust Cocone algorithm accepts the parameters *bbr*, *thif*, *thff* which control which polar balls are used to reconstruct the surface. To test whether Robust Cocone was able to further clean the data points $\mathbf{B}$ obtained from the RkNN method, we ran Robust Cocone with a range of values for *bbr*, *thif*, *thff* in incremental steps. Below we provide the algorithm for the proposed method.

Let $\mathbf{B}$ be an approximation of the boundary points of $\mathbf{P}$ and the parameters for the Robust Cocone algorithm *bbr*, *thif*, *thff* be random values generated by a uniform distribution.

Procedure 2 Hybrid$(\mathbf{P}, k, \alpha)$

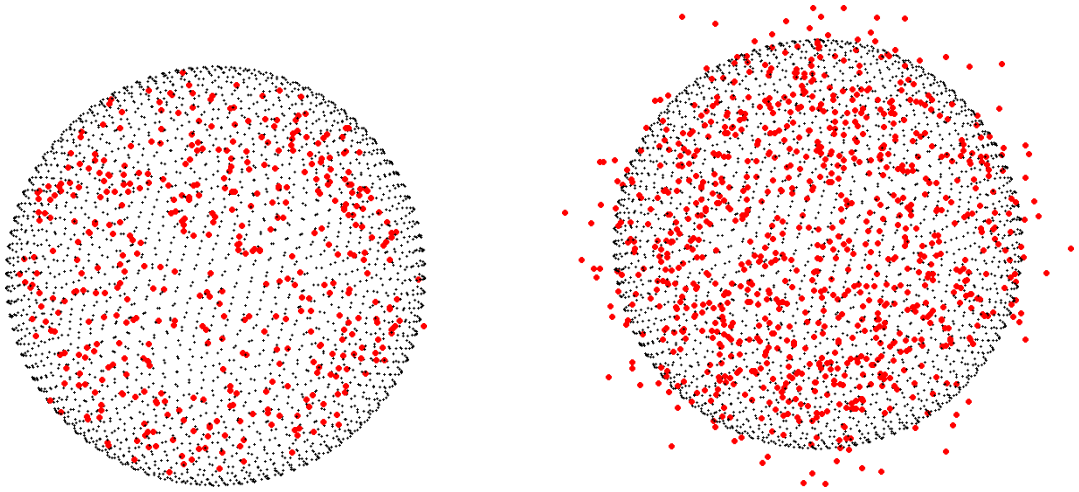
Input: A $m \times n$ matrix $\mathbf{N}, bbf \in [0.15, 0.50], thif \in \{1, 2, \ldots, 10\}, thff \in \{8, 9, \ldots, 20\}$
Output: A matrix $\mathbf{R}$ of points interpolating a subset of the border points of $\mathbf{P}$

```
B ← rkNN(P, k, α)
R ← Robust Cocone(bbf, thif, thff, B)
return R
```

Procedure 2 describes our Robust Cocone approach over a specified range of values for *bbf*, *thif*, *thff*. The authors in [2] specify that these value ranges would produce comparable results which was confirmed in the experimentation.

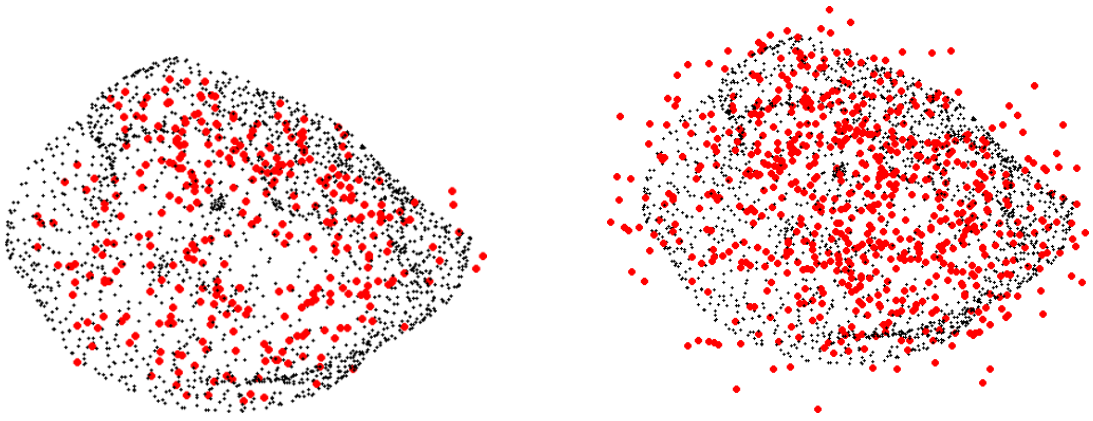
IV. Results

In our experiments, we considered six data sets derived from the tomographic data of three structures: (i) a *sphere* generated in Matlab with uniformly distributed points about the sphere's surface, (ii) tomographic data of a *right medial geniculate*, and (iii) tomographic data of a *suprachiasmatic nucleus*, both obtained from the Japanese Database Center for Life Sciences. In Figs. 3-5 we show the effects of the noise perturbation on each data set for two different noise models (low noise and moderate noise). The clean data points of each object are shown in black while the noisy points are shown in red.



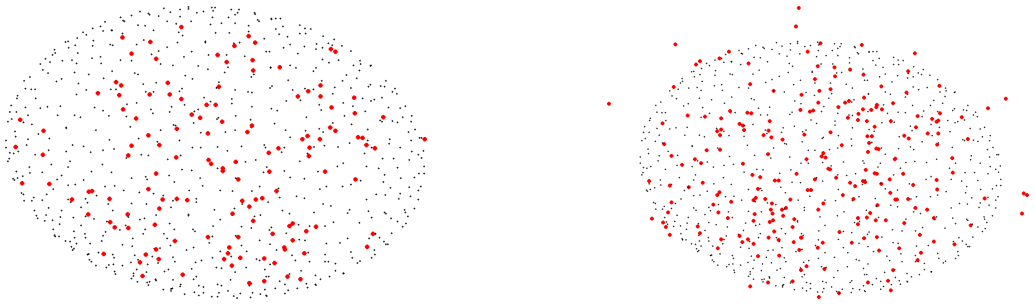
(a) *Low Noise* $\sim$ 14.8% *(added 448 noisy points)* **(b)** *Moderate Noise* $\sim$ 26.7% *(added 932 noisy points)*

Figure 3: *Sphere: Noisy data samples each containing 2562 true surface points*



(a) *Low Noise* $\sim$ 14.5% *(added 332 noisy points)* **(b)** *Moderate Noise* $\sim$ 26.3% *(added 701 noisy points)*

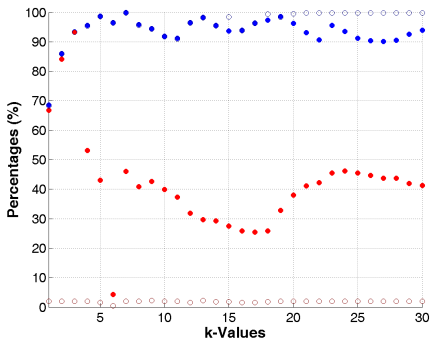
Figure 4: *Right Medial Geniculate: Noisy data samples each containing 1965 true surface points*



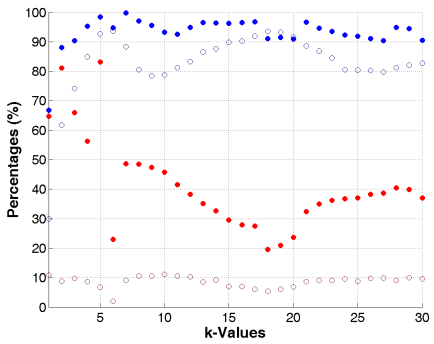
(a) *Low Noise* $\sim$ 16.1% *(added 132 noisy points)* **(b)** *Moderate Noise* $\sim$ 25.2% *(added 231 noisy points)*

Figure 5: *Suprachiasmatic Nucleus: Noisy data samples each containing 686 true surface points*

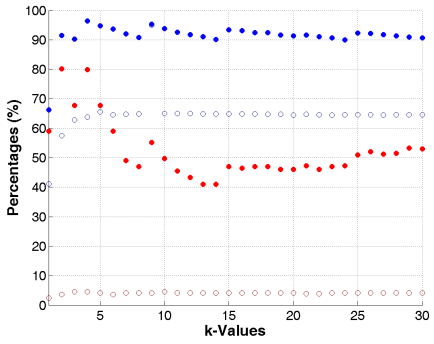
The graphs in Fig. 6 show the minimized percentage of noisy points kept for each k value as α varies in the range $1 \leq \alpha \leq k$, for each of the Methods 1 and 2. For each fixed $1 \leq k \leq 30$, we varied α in the range $1 \leq \alpha \leq k$ and computed the α^* where the minimum percentage of noisy points were left in $\mathbf{B}$. We set a constraint that, for the given minimum α^*, the percentage of clean points kept from the original (noisy) sample be at least 90%. In other words, we were willing to discard at most 10% of the clean data while removing the most amount of noisy points. In Figs. 6(a)-(f), the percentage of remaining clean points are shown in blue, the percentage of remaining noisy points are shown in red. The results for Method 1 are depicted as filled points and the results for Method 2 as hollow points.



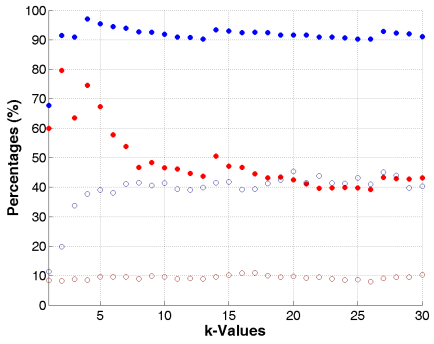
(a) *Sphere* ($\sim$ 14.8% *noise)*



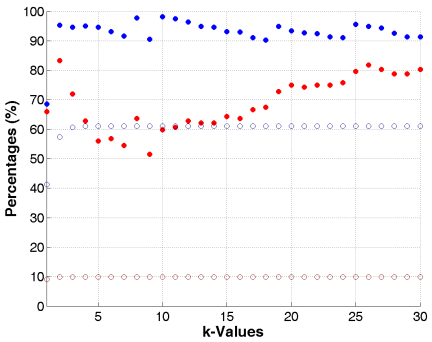
(b) *Sphere* ($\sim$ 26.7% *noise)*



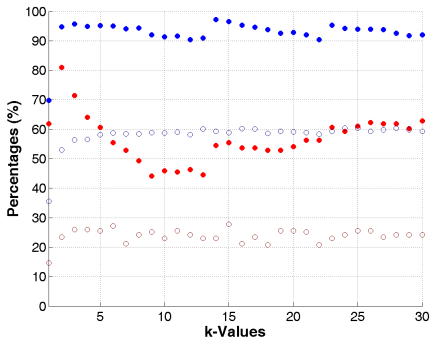
(c) *Right Medial Geniculate* ($\sim$ 14.5% *noise)*



(d) *Right Medial Geniculate* ($\sim$ 26.3% *noise)*



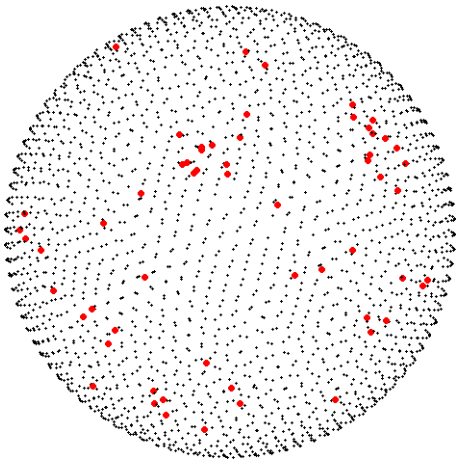
(e) *Suprachiasmatic Nucleus (∼ 16.1% noise)*



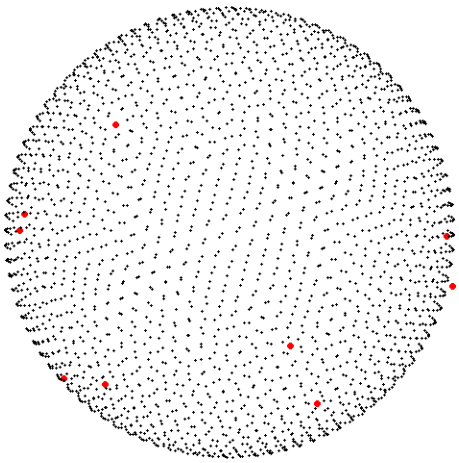
(f) *Suprachiasmatic Nucleus (∼ 25.2% noise)*

Figure 6: *Experimental results comparing the sample point percentages (blue) and noisy point percanteges (red) remaining for Method 1 (filled) and Method 2 (hollow).*

Lastly, the final effect of the proposed methods are shown in Figs. 7-12. The clean sample points remaining are depicted in black while the noise in red.

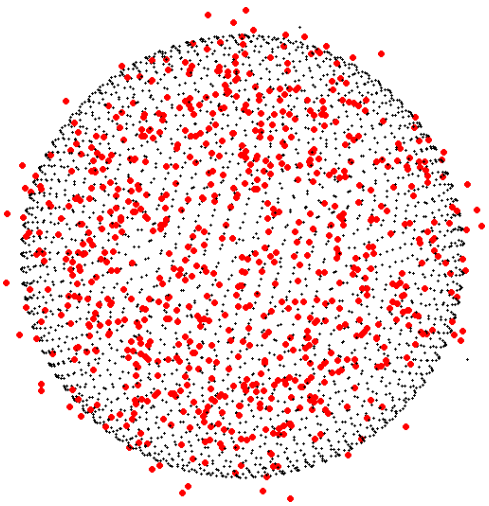


(a) *Method 1: RkNN*

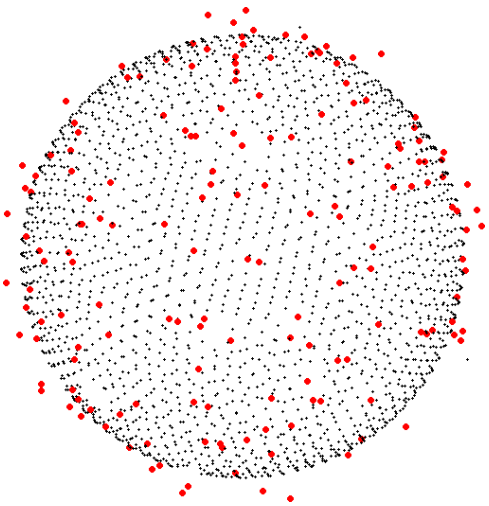


(b) *Method 2: RkNN + Robust Cocone*

Figure 7: *The final effect of our experiment for the Sphere ∼ 14.8% noise dataset.*

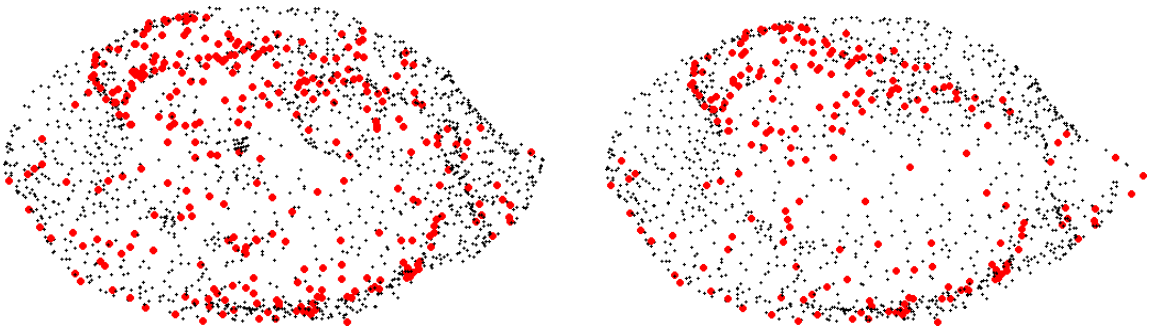


(a) *Method 1: RkNN*



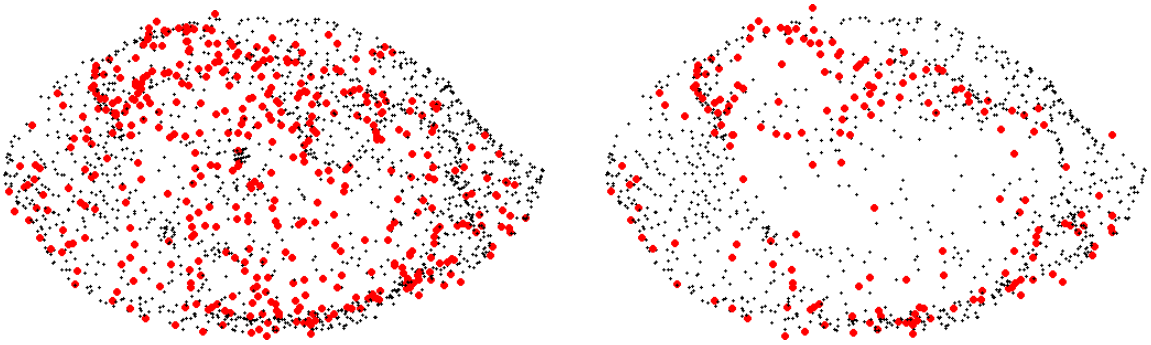
(b) *Method 2: RkNN + Robust Cocone*

Figure 8: *The final effect of our experiment for the Sphere ∼ 26.7% noise dataset.*



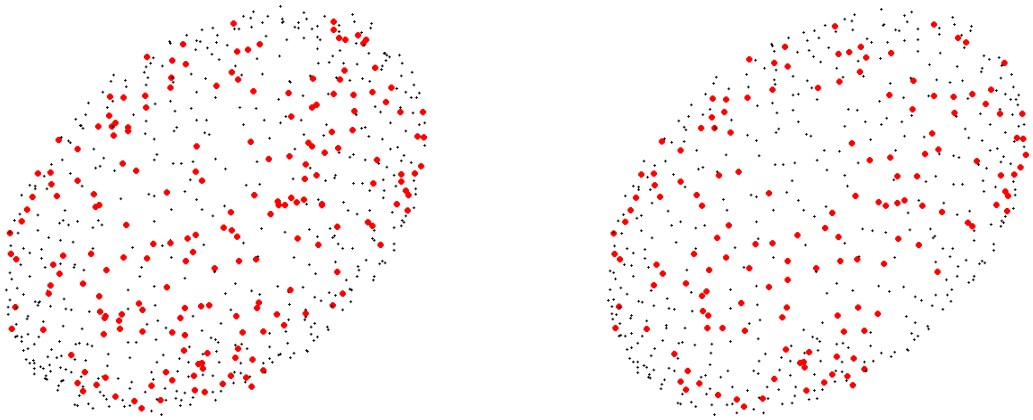
(a) *Method 1: RkNN.* 14.2% *noise remaining.* **(b)** *Method 2: RkNN + Robust Cocone*

Figure 9: *The final effect of our experiment for the Right Medial Geniculate* $\sim$ 14.5% *noise dataset.*



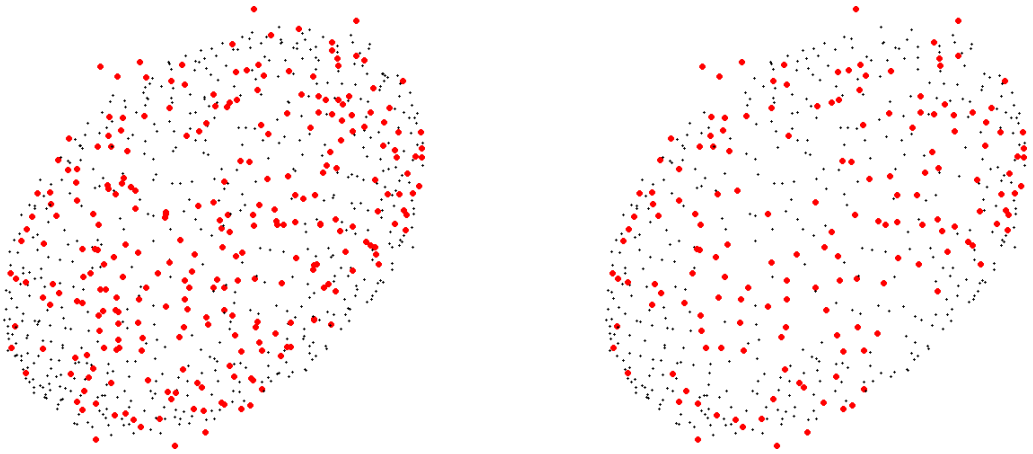
(a) *Method 1: RkNN* **(b)** *Method 2: RkNN + Robust Cocone*

Figure 10: *The final effect of our experiment for the Right Medial Geniculate* $\sim$ 26.3% *noise dataset.*



(a) *Method 1: RkNN* **(b)** *Method 2: RkNN + Robust Cocone*

Figure 11: *The final effect of our experiment for the Suprachiasmatic Nucleus* $\sim$ 16.1% *noise data set.*



(a) *Method 1: RkNN* **(b)** *Method 2: RkNN + Robust Cocone*

Figure 12: *The final effect of our experiment for the Suprachiasmatic Nucleus* $\sim 25.2\%$ *noise dataset.*

V. Discussion

To test our two proposed methods, we considered six datasets derived from three tomographic models over $k \in \{1,\ldots,30\}$ and $\alpha \in \{1,\ldots,k\}$. When $k > 30$ our methods no longer effectively removed noise from our data and in some cases, retained a greater percentage of noise than clean sample data points. Our first proposed method which utilizes the RkNN algorithm showed extreme promise by removing anywhere between 40% to 60% of the noise while preserving at least 90% of the sample points and performed consistently even in the presence of large quantities of noise. Our second method which utilizes the RkNN and Robust Cocone algorithms was particularly effective in removing large quantities of noise from the tomographic data; however, when the sample size was reduced in conjunction with an increase of noise, this method did not preserve a sufficiently large percentage of sample points which may call into the question the structure of the tomographic model.

In Fig. 6a, Method 1 successfully removed 74.55% of the noise while preserving 96.37% of the clean data for a (k,α) value of $(17,16)$. Method 2 showed a marked improvement over Method 1 by consistently removing 97.99% of the noise while preserving 99.69% of the clean data. Increasing the percentage of noise by 12 points, Fig. 6b. shows both methods are resistant to the increase in noise.

Reducing the number of sample points while preserving similar levels of noise, Fig. 6c showed a marked decline in performance for both methods. Method 1 successfully removed 51.04% of the noise while preserving 90.13% of the clean data for a (k,α) value of $(14,11)$. Method 2 was still highly effect as a method for noise removal consistently removing 95.78% of the noise; however, the method consistently preserved only 64.83% of the clean data. Though the disparity between these two values is still significantly large, and nearly all noise is still being removed, a substantial loss of clean data will encourage the loss of key features of the original tomographic model. Fig. 6d shows Method 1 is still particularly resistant to changes in noise while Method 2 shows a further drop in performance by consistently removing 90% of the noise and preserving only 40% of the clean data.

Lastly, we once again significantly reduce the sample point count while preserving similar levels of noise. Fig. 6e showed a similar marked decline in performance for both methods. Method 1 successfully removed 48.48% of the noise while preserving

98.10% of the clean data for a (k, α) value of $(10, 8)$. Method 2 still consistently removes 89.39% of the noise; however, it now only preserves 63.41% of the clean data. Fig. 6f, much like the other figures, shows Method 1 is still resistant to changes in noise while Method 2 again shows a drop in performance.

For either of our methods, any (k, α) value within the proposed domain significantly reduced the percentage of remaining noise while preserving a sufficiently large percentage of the original tomographic data. Method 2 is recommended when the tomographic data features a large number of observations and can provide some guarantees to the level of noise within the data for any (k, α) value in the domain. Otherwise, Method 1 is recommended for (k, α) values where $k \in \{10, \ldots, 20\}$ and $\alpha \in \{\frac{1}{2}k, \frac{1}{2}k + 1, \ldots, k\}$.

VI. Conclusion

In this paper, we considered two discrete tomographic models to reduce the influence of noise on tomographic data while preserving the clean data as much as possible. The experiment was quantified by the minimization of the percent of remaining noise while maintaining a threshold value of 90% of the clean data. Both methods proved to be very useful in mitigating the presence of noise which will allow us to accurately determine the boundary points of the tomographic data. The first method is particularly resilient to fluctuations in the size of our clean data and changes in noise, while the second method is very effective at masking the presence of noise for sufficiently large sizes of the clean data.

References

[1] Nina Amenta, Sunghee Choi, and Ravi Krishna Kolluri. The power crust. In *Proceedings of the sixth ACM symposium on Solid modeling and applications*, pages 249–266. ACM, 2001.

[2] Tamal K Dey and Samrat Goswami. Provable surface reconstruction from noisy samples. In *Proceedings of the twentieth annual symposium on Computational geometry*, pages 330–339. ACM, 2004.

[3] Inna Tishchenko. Surface reconstruction from point clouds.

[4] Chenyi Xia, Wynne Hsu, Mong Li Lee, and Beng Chin Ooi. Border: efficient computation of boundary points. *Knowledge and Data Engineering, IEEE Transactions on*, 18(3):289–303, 2006.