1

Straightening 3-D surface scans of curved natural history specimens for taxonomic research

James Church*[1], Ray Schmidt†[2], Henry Bart Jr.‡[2], Xin Dang§[3] and Yixin Chen¶[1]

[1]Department of Computer and Information Science, University of Mississippi, University, MS 38677, USA.
[2]Department of Ecology and Evolutionary Biology, Tulane University, New Orleans, LA 70118, Tulane University Biodiversity Research Institute, Belle Chasse, LA 70037, USA
[3]Department of Mathematics, University of Mississippi, University, MS 38677, USA.

Summary. Two challenges for taxonomists are proper identification of specimens to known species and extracting information from specimens to diagnose new species. Both tasks are complicated by the very large numbers of known and unknown species and the dwindling numbers of qualified taxonomists to identify/diagnose them all. Automated species identification is a tool that can assist taxonomists facing this challenge. This paper looks at one aspect of automated species identification: unfolding curved specimens, which commonly occurs when specimens are prepared for storage in natural history collections. Here we attempt to address the rather extreme case of an elongate fish specimen coiled along its medial axis. The medial axis is the set of all points within an object with the shortest distance to at least two different points on that object's surface, where "distance" (typically Euclidean) is determined by the application. Medial Axis Estimation is a challenging problem that arises when the surface itself is sampled (i.e. incomplete). In this paper, we look at various techniques for estimating the medial axis of an object, then we propose a new method for medial axis estimation based on localized spatial depth. We extend the idea of localized spatial depth-based medial axis further by applying an original ridge detector. We conclude with a comparison of our approach with The Power Crust approach using artificial data.

Key words: medial axis, ridge detection, taxonomic research

* jcchurch@go.olemiss.edu
† rschmidt@tulane.edu
‡ hbartjr@tulane.edu
§ xdang@olemiss.edu
¶ ychen@cs.olemiss.edu

1.1 Introduction

Taxonomy is a field of biological study in which specimens are classified in groups based on unique characteristics that members of each group share in common and unique names are assigned to identify each group. The field of taxonomy is confronted with several challenges [1] [2]. First, many parts of the world are unexplored by taxonomists and some of these areas are rich with undiscovered species. Second, the pace of taxonomic discovery, as traditionally practiced, has been slow, and the number of practicing taxonomists has been in decline for several decades, resulting in what has been termed the *taxonomic impediment*. Third, human destruction of natural habitats, especially in species rich areas, has resulted in a *biodiversity crisis*, and it is feared that many species will go extinct before they can be discovered and described. Computer tools have the potential to assist taxonomists by automating and expediting the process of diagnosing specimens as members of either known or unknown species [3] [4]. In one computer-aided approach, a specimen is scanned using a 3D scanner, the scan is digitally landmarked, geometric features are extracted from the landmarks and analyzed using a heuristic function. If the heuristic function fails to identify the specimen as a member of a known species with a certain degree of confidence, it is possible that the specimen is representative of a new species. The specimens used in taxonomic studies are typically preserved specimens obtained from natural history museums. Often natural history specimens are preserved with different degrees of curvature of their bodies. Elongate specimens such as eels preserved in jars often take the highly curved shape of their containers, making it difficult to extract features from 3-D scans of these specimens. This paper addresses the challenge of straightening curved natural history specimens. Our goal is to estimate the medial axis of a 3-D scan of an elongated fish specimen for the purposes of unfolding the specimen.

The medial axis is the set of all points within an object with the shortest distance to at least two different points on that object's surface, where "distance" (typically Euclidean) is determined by the application. It is an important tool in computer vision applications in order to determine the "skeleton" of a shape or to approximate surface reconstruction. Applications of the medial axis range from object unfolding to automatic rigid skeleton formation intended for physics applications. In this paper, we focus on medial axis estimation with the given assumption that our input shape consists of only sample points along the surface and that holes exist. The surface scans are often created using 3D scanners and the problems related to scanners are discussed by Bajaj et al. [5].

1.1.1 Medial Axis Estimators

The medial axis was first proposed by Blum [6] using an idea that he called the "grass fire" analogy (now called the "prarie fire" analogy): if a fire is lit along the perimeter of a shape, it should burn inward, leaving the internal skeleton of the object at the points where the fire is quenched. Boissonnat [7] first proposed that the medial axis of a shape can be determined using the set of points produced by a Voronoi diagram that

also lie on the interior of that shape. Edelsbrunner and Mücke [8] took Boissonnat's ideas and gave them a more formal definition using the α-shape algorithm. Each of these approaches assume that a complete knowledge of the shape exists.

Often a perfect knowledge of an object's surface does not exist. The scanning process of a object is sampled from different vantage points and noise is introduced, leaving the object incomplete. For this reason, medial axis estimation techniques were developed. The relevant literature focuses on Delaunay or Voronoi decomposition of points in order to reconstruct the medial axis and surface reconstruction. Amenta et al. [9] created The Power Crust which uses the Voronoi decomposition and focuses on narrow structures in shapes that lead to confusion as to which points are "interior" or "exterior". The Tight Cocone algorithm developed by Dey and Goswami [10] uses the Delaunay triangulation in three dimensions to create a convex hull of the shape. Then the algorithm removes tetrahedrons that lie on the exterior of the shape, leaving an approximation of the surface. The λ-medial axis by Chazal and Lieutier [11] is an attempt at approximating the medial axis of a noisy set of surface points by first sampling the points and then finding the internal set of Delaunay points.

1.1.2 Ridge Detectors

Ridge detectors are an element of computer vision that help to simplify the analysis of images so that ridges may be applied to unique applications. Most importantly, we focus on the work of Lindeberg [12] describing edge and ridge detection in two dimensions. While there is some work on extending the work of Lindeberg in three dimensions [13], it appears that previous attempts at this use ad hoc methods for formulating a 3D ridge. What we present is a complete work on extending ridge detection to three dimensional images and we set ourselves up for future work in ridge detection beyond three dimensions.

Ridge and edge detection are closely related. Canny [14] sets out the two criteria for an edge in two dimensions. First, the edge detector should be robust with respect to noise. Second, the edge detector should be accurate in order to localize each edge. We apply the same criteria to a ridge detector in three dimensions.

Traditionally a ridge is defined in two dimensions as a raised separator (e.g. a mountain range) which divides two distinct regions. The ridge itself extends in two directions orthogonal to the two regions. Describing the property of a ridge in three dimensions requires us to think abstractly. We define a ridge as a raised separator which divides an area along two directions which are orthogonal to each other. A third direction orthogonal to both of the two directions represents the direction of the ridge itself. In an n-dimensional setting, we define a ridge as a raised separator which divides an area along n-1 orthogonal directions. The nth-direction (also orthogonal) is the direction of the ridge. Based on this definition, we can set the following criteria that must be met for a ridge: (1) There should be a negative gradient change in two orthogonal directions from a point; (2) In the direction orthogonal to this gradient change, there should be no significant gradient change; (3) This should be invariant to scale; (4) This should be robust with respect to noise.

1.1.3 Paper Organization

The remainder of this work is organized as follows. In section 1.2 we introduce the rational for using the localized spatial depth rather than the Voronoi decomposition. In section 1.3 we provide a formal definition for a ridge in three dimensions. In section 1.4 we outline an approach to unify chains of ridges. In section 1.5 we provide a complete algorithm for our approach to medial axis estimation and ridge alignment. In section 1.6 we provide a comparison of our approach with artificial data sets and compare them to ground truth as well as show examples our approach on using real data. In section 1.7 we discuss potential open questions introduced by our research.

1.2 The Spatial Depth Formulation

Unlike other contemporary work, this approach to medial axis estimation does not utilize the Delaunay or Voronoi decomposition. Rather, we use a localized spatial median [15] approach to estimating the medial axis. The advantage of the Voronoi decomposition is that it always provides the discrete medial axis, provided that the shape is perfectly known. The various techniques involving the Voronoi decomposition and sampled input data focus on different aspects of the problem of incompleteness. Our approach using the localized spatial depth formalization allows us to create an implicit interpolation of the medial axis. Areas without sufficient coverage that fail to create a ridge are thus passed over by the algorithm. Our reason for opting for the spatial median formulation is that it turns the discrete set of surface points into a suitable image which then allows us to take advantage of an array of computer vision techniques.

The initial input is the sampled surface points of an object. We begin by crafting a localized spatial depth image of the initial shape at a specified grid interval. The spatial depth of a surface scan is created by calculating a weight for each point p within an image based on a single observation (or surface scan point) o at a particular σ window size:

$$w(p,o) = e^{-\frac{(p-o)^2}{\sigma^2}}.$$

Once the weights have been computed at point p, we combine those weights of all surface scan points O into a spatial depth:

$$Spatial\,Depth(p,O) = 1 - \left\| \sum_j^n \left[\frac{w(p,O_j)}{\sum_{i=1}^n w(p,O_i)} \times \frac{p-O_j}{\sqrt{\sum_{i=1}^n (p-O_i)^2}} \right] \right\|.$$

The spatial depth is a reflection of the symmetry of a point in space to the surface points of our scan. The Spatial Depth formula has a bound of 0 to 1 and higher values indicate a better symmetry of surrounding surface points. It is by this property that we formulate our ridge detector.

1.2.1 Selecting the Right σ

Care should be taken in selecting the proper σ value in order to reconstruct the medial axis when using the spatial depth formulation in the context of estimating the the medial axis of a shape. Values of σ that are too small run the risk of being too local for a proper analysis of the ridges. Values of σ that are too large run the risk of creating a border area where the calculated values within σ units of the image edge are ineffective due to edge noise.

We created a cylinder shape to use as a test case for properly testing the most effective value of σ. The shape consists of a series of rings stacked to resemble that of a three dimensional cylinder. Each ring in the cylinder consist of 128 evenly spaced points along the perimeter of a circle with a radius of 1. The shape consists of 24 rings that are evenly spaced $\frac{1}{6}$ units apart. The overall shape consists of 3072 surface points. The overall dimension of the shape is 2 units by 2 units by 3.833 units.

A sufficiently small delta of 0.005 was selected and a spatial depth image was created using 100 σ values ranging from 0.01 to 2. The small delta produced an image dimension of 401 by 401 by 767 voxels. Only the center line of the cylinder image was relevant to our analysis. We plucked the single dimension line existing long the z-axis coordinate for all 100 σ values. Since the center line of the shape represents the highest conceptual spatial depth, we took the sum of each line. Our finding for this circumstance was that the σ representing the highest summation of spatial depths was 0.5, which is $\frac{1}{4}$ the diameter of our cylinder.

In practice, we feel that a σ should be selected that is roughly $\frac{1}{4}$ the diameter of tubular shape being studied. It may require observing the various appendages of the shape prior to analysis and determining a proper σ value.

1.3 Ridge Detection in 3D Space

In order to determine the ridges in three dimensional space, we must rotate each point within the (x,y,z) coordinate frame to a (p,q,r) coordinate frame individually so that one of the three directions is aligned with the ridge direction. To reduce the amount of noise in the image, a scale-space filter is first applied to the image. Scale space smoothing is a smoothing function using parameter t to determine the smoothness. t is a non-negative integer value which represents the scale of the smoothing. A larger t indicates a greater smoothing scale. If $t = 0$, the image is not smoothed. For an input samples with no noise, no smoothing is required. For noisy data, larger values of t will be required. The smoothing function is

$$g(x,y,z;t) = \frac{e^{\frac{-(x^2+y^2+z^2)}{2t}}}{2\pi t^{\frac{3}{2}}}.$$

The smoothed image is defined by the convolution of the smoothing function and the image function (a.k.a the brightness function):

$$L(x,y,z;t) = g(x,y,z;t) * f(x,y,z).$$

Lindeberg's 2D formulation of the scale space ridge detector required that each voxel within the image be rotated from a (x,y) coordinate frame to a (p,q) coordinate frame. The rotation matrix used to rotate the points was based on the eigenvectors discovered by the eigendecomposition of the Hessian matrix at point (x,y). In order to rotate each point within the image from the (x,y,z) coordinate frame to the (p,q,r) coordinate frame, we compute the 1^{st} and 2^{nd} derivatives of the image at point (x,y,z). The 1^{st} derivatives are known as L_x, L_y, and L_z. From the 1^{st}, we find the 2^{nd} derivatives by taking the derivatives of each of the 1^{st} derivatives. The 2^{nd} derivatives are known as L_{xx}, L_{xy}, L_{xz}, L_{yy}, L_{yz}, and L_{zz}.

Using the 2^{nd} derivatives of the image, we formulate the Hessian Matrix:

$$H_{(x,y,z)} = \begin{bmatrix} L_{xx} & L_{xy} & L_{xz} \\ L_{xy} & L_{yy} & L_{yz} \\ L_{xz} & L_{yz} & L_{zz} \end{bmatrix}.$$

Using the Hessian matrix, we perform the eigendecomposition to discover the eigenvalues of $H_{(x,y,z)}$: $\lambda_1, \lambda_2, \lambda_3$. The eigenvalues correspond to the 2^{nd} derivatives of the system when rotated to the p, q, and r axis:

$$\begin{bmatrix} \lambda_1 \\ \lambda_2 \\ \lambda_3 \end{bmatrix} = \begin{bmatrix} L_{pp} \\ L_{qq} \\ L_{rr} \end{bmatrix}^T.$$

We also determine the eigenvectors for each of the eigenvalues,

$$V = \begin{bmatrix} x_1 & x_2 & x_3 \\ y_1 & y_2 & y_3 \\ z_1 & z_2 & z_3 \end{bmatrix},$$

where each column of V represents an eigenvector of the Hessian matrix.

We wish to formulate a rotation matrix that will transform the 1^{st} derivatives of the image along the x, y, and z axis to the p, q, and r axis. We call this 3×3 rotation matrix R. If ∇g is the 1^{st} derivatives at a point in (x,y,z) space, then $R \times \nabla g$ equals our 1^{st} derivatives in (p,q,r) space, i.e.

$$\nabla g = \begin{bmatrix} \partial x \\ \partial y \\ \partial z \end{bmatrix},$$

$$R \times \nabla g = \begin{bmatrix} \partial p \\ \partial q \\ \partial r \end{bmatrix}.$$

We can show that

$$
\begin{aligned}
&(R \times \nabla g) \times (R \times \nabla g)^T \times L \\
&= \begin{bmatrix} L_{pp} & L_{pq} & L_{pr} \\ L_{pq} & L_{qq} & L_{qr} \\ L_{pr} & L_{qr} & L_{rr} \end{bmatrix} \\
&= \begin{bmatrix} \partial p \partial pL & \partial p \partial qL & \partial p \partial rL \\ \partial p \partial qL & \partial q \partial qL & \partial q \partial rL \\ \partial p \partial rL & \partial q \partial rL & \partial r \partial rL \end{bmatrix} \\
&= \begin{bmatrix} \lambda_1 & 0 & 0 \\ 0 & \lambda_2 & 0 \\ 0 & 0 & \lambda_3 \end{bmatrix}.
\end{aligned}
$$

Now we evaluate $(R \times \nabla g) \times (R \times \nabla g)^T \times L$:

$$
\begin{aligned}
&(R \times \nabla g) \times (R \times \nabla g)^T \times L \\
&= (R \times \nabla g \times \nabla g^T \times R^T) \times L \\
&= R \times (\nabla g \times \nabla g^T \times L) \times R^T \\
&= R \times H_{(x,y,z)} \times R^T
\end{aligned}
$$

By definition,

$$R \times H_{(x,y,z)} \times R^T = V^T \times H_{(x,y,z)} \times V.$$

Therefore, the matrix V^T is our rotation matrix R. With this we can determine L_p, L_q, L_r:

$$R \times \begin{bmatrix} L_x L_y L_z \end{bmatrix}^T = \begin{bmatrix} L_p L_q L_r \end{bmatrix}^T.$$

Using our values of $L_p, L_q, L_r, L_{pp}, L_{qq}$, and L_{rr}, we can define a ridge. A ridge in three dimensions is a point in space which has a negative gradient change along two 2^{nd} derivatives and no negative gradient change along the third 2^{nd} derivative. It's this third direction where a ridge line exists that can be followed for our purposes. We can define a ridge using the following:

$$
\begin{cases} L_p = 0 \\ L_{qq} < 0 \\ L_{rr} < 0 \\ |L_{pp}| < |L_{qq}| \\ |L_{pp}| < |L_{rr}| \end{cases} or \begin{cases} L_q = 0 \\ L_{pp} < 0 \\ L_{rr} < 0 \\ |L_{qq}| < |L_{pp}| \\ |L_{qq}| < |L_{rr}| \end{cases} or \begin{cases} L_r = 0 \\ L_{pp} < 0 \\ L_{qq} < 0 \\ |L_{rr}| < |L_{pp}| \\ |L_{rr}| < |L_{qq}| \end{cases}
$$

1.4 Implementing Canny's Edge Detector in 3D

To assist in determining which points can be identified as ridges, we implement a "dual threshold" technique for identifying ridges similar to the approach developed

by Canny for identifying edges. Canny's edge detector required two things to be effective: strength and direction.

We formulate the strength of the ridge by summing the absolute values of the 2^{nd} derivatives orthogonal to that of the ridge direction. Due to inevitable nuisances (i.e. rounding error), we must apply a narrow threshold on the ridge conditionals. We have found a threshold of 0.001 to be effective. We have developed a "ridge strength" (*RS*) to help us determine the strength of an edge based on the eigengap of a ridge in pqr-space:

$$RS = \begin{cases} |L_{rr}| + |L_{qq}| \\ \qquad \text{if } |L_p| < ridgeThresh \wedge \\ \qquad\quad L_{qq} < 0 \wedge L_{rr} < 0 \wedge \\ \qquad\quad |L_{pp}| < |L_{qq}| \wedge |L_{pp}| < |L_{rr}| \\ |L_{pp}| + |L_{rr}| \\ \qquad \text{if } |L_q| < ridgeThresh \wedge \\ \qquad\quad L_{pp} < 0 \wedge L_{rr} < 0 \wedge \\ \qquad\quad |L_{qq}| < |L_{pp}| \wedge |L_{qq}| < |L_{rr}| \\ |L_{pp}| + |L_{qq}| \\ \qquad \text{if } |L_r| < ridgeThresh \wedge \\ \qquad\quad L_{pp} < 0 \wedge L_{qq} < 0 \wedge \\ \qquad\quad |L_{rr}| < |L_{qq}| \wedge |L_{rr}| < |L_{pp}| \\ 0 \\ \qquad \text{otherwise} \end{cases}.$$

Provided that the *RS* score at a point is greater than 0, we determine the ridge direction *RD* by retrieving the eigenvector of the primary ridge determined by the maximum of L_{pp}, L_{qq}, or L_{rr}.

$$RD = \begin{cases} \begin{bmatrix} x_1 y_1 z_1 \end{bmatrix}^T, & \text{if } L_{pp} = max(L_{pp}, L_{qq}, L_{rr}), \\ \begin{bmatrix} x_2 y_2 z_2 \end{bmatrix}^T, & \text{if } L_{qq} = max(L_{pp}, L_{qq}, L_{rr}), \\ \begin{bmatrix} x_3 y_3 z_3 \end{bmatrix}^T, & \text{if } L_{rr} = max(L_{pp}, L_{qq}, L_{rr}). \end{cases}$$

It should be noted that while the input shape can and will contain holes, our algorithm makes no guarantees as to whether or not a discovered ridge exists on the interior or exterior of the shape. Still we can reduce the number of exterior ridges by filtering out all of the candidate ridges if they do not also have a local spatial depth greater than a high threshold (0.9 in our experiments).

Once we have determined the set of candidate ridges, their scores and their directions, we must now select two threshold values by which to select our ridges. We call these thresholds *high* and *low*. First, the image is scanned for all voxels with *RS* scores greater than *high*. These scores and locations are added to a priority queue where priority is determined by greatest *RS* score. While the priority queue is not

empty, take the front element off the queue and label the point as a confirmed point. We then look at the two voxels in the directions of RD and $-RD$ and determine if those voxels have a RS score greater than low. If so, we add the voxel location and RS to the priority queue.

1.5 The Complete Algorithm

1 Create a localized spatial depth image I of the shape using a predetermined σ window size.
2 Perform the scale space smoothing filter on the image I with preferred scaling factor t.
3 For each point within the image I, determine the 1^{st} which represents L_x, L_y, and L_z.
4 For each of the 1^{st} derivatives, determine the 2^{st} derivatives and formulate the Hessian matrix H.
5 Compute $[\lambda, V] = eig(H)$, where λ and V are the eigenvalues and eigenvectors of matrix H. The λ represents the 2^{nd} derivative directions in the (p,q,r)-coordinate frame: L_{pp}, L_{qq}, L_{rr}.
6 Compute the rotation matrix R for this point, which is V^T.
7 Determine the 1^{st} derivative directions in the (p,q,r)-coordinate frame, which is $R \times [L_x, L_y, L_z]^T$, which are represented by L_p, L_q, L_r.
8 For each point that meets the criteria for a ridge, compute the ridge score for this ridge point.
9 For each point that meets the criteria for a ridge, note the eigenvector associated with the largest eigenvalue of the Hessian matrix at this point. This becomes the dominate eigenvector.
10 Select a desired low and $high$ threshold values for ridge points.
11 Create a three dimensional matrix equal to the size of the image I composed of all Boolean false flags. This matrix will represent the confirmed ridges.
12 For each ridge point that has a RS greater than the $high$ threshold, add that ridge point's ridge score and location to the end of a priority queue.
13 While the priority queue is not empty:
   a Sort the priority queue by the ridge score in descending order.
   b Pop the first element from the priority queue. Flag this point in the confirmed ridge point matrix as true. If it has been previously flagged as a confirmed ridge, skip the remaining steps in this loop iteration.
   c Test the neighboring point in the direction of the dominate vector for this point. If the neighboring point meets the criteria for a ridge and has a ridge score greater than the low threshold, add the neighboring ridge score and the ridge point location to the end of the priority queue.
   d Test the neighboring point in the direction opposite of the dominate vector for this point. If the neighboring point meets the criteria for a ridge and has a ridge score greater than the low threshold, add the neighboring ridge score and the ridge point location to the end of the priority queue.

1.6 Experimental Result

In order to test the validity of our algorithm, we created three dimensional images representing both artificial structures and real objects scanned with a 3D scanner. For each shape, we select an appropriate σ, smoothing parameter t, and ridge thresholds *high* and *low* and executed the algorithm. Examples of artificial images can been seen in Figures 1.1 and 1.2 and a surface image of an Anguilla rostrata (American Eel) in Figure 1.3. For the artificial structures, we evaluate each image based on the identified ridges by how closely they conform to a known center line of the shape. We have compared our results with the Power Crust method using a simple correspondence algorithm using a Root-Mean-Squared Error formulation comparing each point in the discovered medial axis to the nearest point in the ground truth (RMSE MA) and comparing each point the ground truth to the nearest point in the discovered medial axis (RMSE Truth). We can show that the two approaches are comparable in Table 1.1.

U-Pipe	RMSE MA	RMSE Truth	Avg.
L. Spatial Depth	0.0786	0.2088	0.1437
Power Crust	0.0042	0.1117	0.0579
Cork Screw	RMSE MA	RMSE Truth	Avg.
L. Spatial Depth	1183.4	6748.3	3965.9
Power Crust	16182.0	2564.0	9373.1

Table 1.1: We show that the Localized Spatial Depth and the Power Crust method have comparable results.

1.7 Conclusions

In each experimental result, we have discovered enough ridge lines in order to connect the ridges using Dijkstra's shortest path algorithm. Our data sets mostly represent the elongated structures of fish with an anterior, a narrow body, and a posterior. By flagging the two points representing the anterior and posterior, we can use this technique to create a full length vertebra of the internal structure of the shape. After obtaining the body-length vertebra, to unfold the structure is trivial: associate each surface point with the nearest vertebra point and align it on a single axis, taking care to rotate the point with respect to the vertebrae point immediately preceding the connecting vertebra point. There is one problem with this approach which needs addressing in our future work. The body contortions of the fish specimen will cause the fish's body to twist around its own internal structure. The end result of our unfolding process will also be a twisted contortion around the discovered vertebra. We feel that the one way to alleviate this issue is to add landmarks along the dorsal side of the

fish specimen. With this information, we can perform a rotation along the roll axis of the specimen.

Our algorithm depends heavily on the selection of a σ window. A good σ window can fall into a range of values, but it is still dependent on the diameter of the appendages of the object being analyzed. We feel that there is more research that can be done into evaluating the shape prior to the start of the algorithm that could be used in σ discovery.

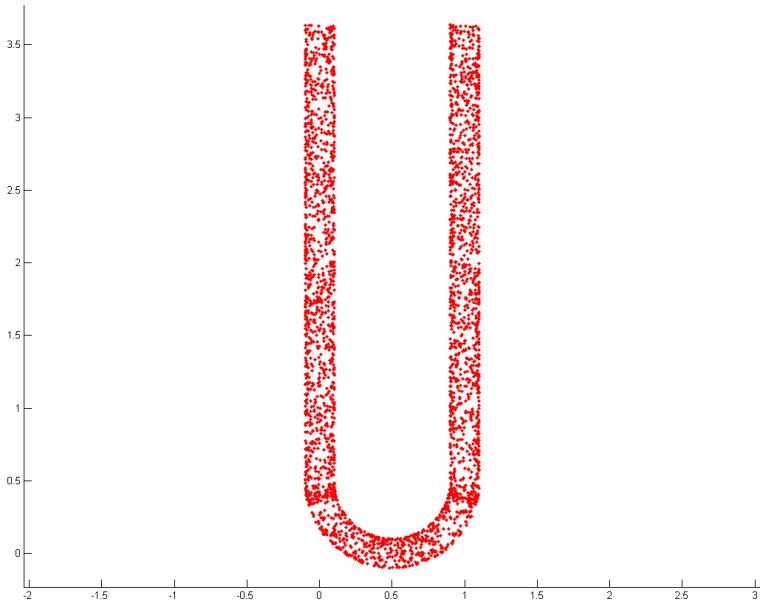
Acknowledgment

This material is based upon work supported by the National Science Foundation under Grant No. NSF MCB-1027989 and MCB-1027830. We would like to thank Aishat Aloba for processing large data sets.

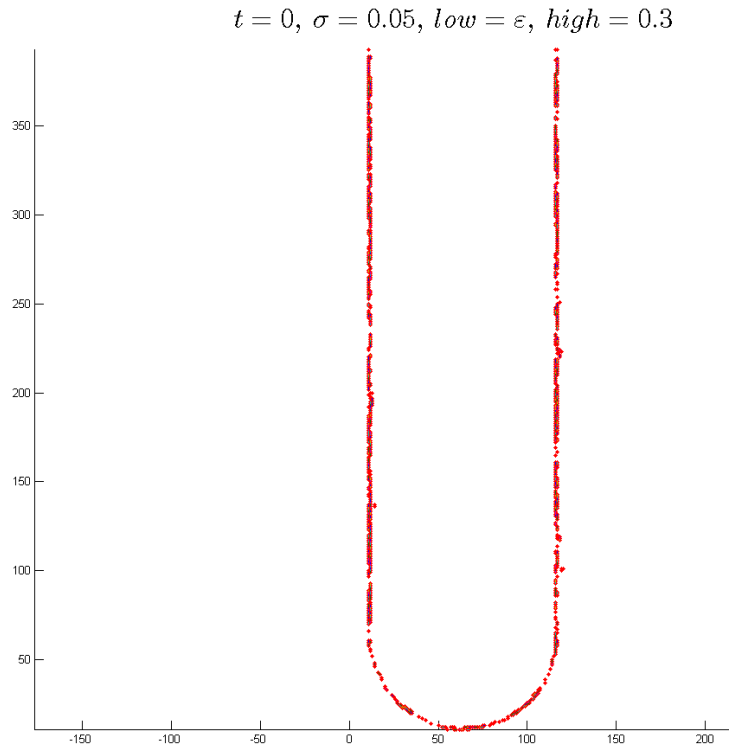
References

1. Y. Chen, H. Bart, X. Dang, and H. Peng, "Depth-based novelty detection and its application to taxonomic research," in *Data Mining, 2007. ICDM 2007. Seventh IEEE International Conference on*. IEEE, 2007, pp. 113–122.
2. Y. Chen, H. L. Bart Jr, S. Huang, and H. Chen, "A computational framework for taxonomic research: diagnosing body shape within fish species complexes," in *Data Mining, Fifth IEEE International Conference on*. IEEE, 2005, pp. 4–pp.
3. M. R. de Carvalho, F. A. Bockmann, D. S. Amorim, C. R. F. Brandão, M. de Vivo, J. L. de Figueiredo, H. A. Britski, M. C. de Pinna, N. A. Menezes, F. P. Marques *et al.*, "Taxonomic impediment or impediment to taxonomy? a commentary on systematics and the cybertaxonomic-automation paradigm," *Evolutionary Biology*, vol. 34, no. 3, pp. 140–143, 2007.
4. J. E. Rodman and J. H. Cody, "The taxonomic impediment overcome: Nsf's partnerships for enhancing expertise in taxonomy (peet) as a model," *Systematic Biology*, vol. 52, no. 3, pp. 428–435, 2003.
5. C. L. Bajaj, F. Bernardini, and G. Xu, "Automatic reconstruction of surfaces and scalar fields from 3d scans," in *Proceedings of the 22nd annual conference on Computer graphics and interactive techniques*. ACM, 1995, pp. 109–118.
6. H. Blum, "A transformation for extracting new descriptors of shape," *Models for the perception of speech and visual form*, vol. 19, no. 5, pp. 362–380, 1967.
7. J.-D. Boissonnat, "Geometric structures for three-dimensional shape representation," *ACM Transactions on Graphics (TOG)*, vol. 3, no. 4, pp. 266–286, 1984.
8. H. Edelsbrunner and E. P. Mücke, "Three-dimensional alpha shapes," *ACM Transactions on Graphics (TOG)*, vol. 13, no. 1, pp. 43–72, 1994.
9. N. Amenta, S. Choi, and R. K. Kolluri, "The power crust," in *Proceedings of the sixth ACM symposium on Solid modeling and applications*. ACM, 2001, pp. 249–266.
10. T. K. Dey and S. Goswami, "Tight cocone: a water-tight surface reconstructor," in *Proceedings of the eighth ACM symposium on Solid modeling and applications*. ACM, 2003, pp. 127–134.
11. F. Chazal and A. Lieutier, "The "λ-medial axis"," *Graphical Models*, vol. 67, no. 4, pp. 304–331, 2005.

12. T. Lindeberg, "Edge detection and ridge detection with automatic scale selection," *International Journal of Computer Vision*, vol. 30, no. 2, pp. 117–156, 1998.
13. Y. Sato, S. Nakajima, N. Shiraga, H. Atsumi, S. Yoshida, T. Koller, G. Gerig, and R. Kikinis, "Three-dimensional multi-scale line filter for segmentation and visualization of curvilinear structures in medical images," *Medical image analysis*, vol. 2, no. 2, pp. 143–168, 1998.
14. J. Canny, "A computational approach to edge detection," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, no. 6, pp. 679–698, 1986.
15. Y. Chen, X. Dang, H. Peng, and H. L. Bart, "Outlier detection with the kernelized spatial depth function," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 31, no. 2, pp. 288–305, 2009.

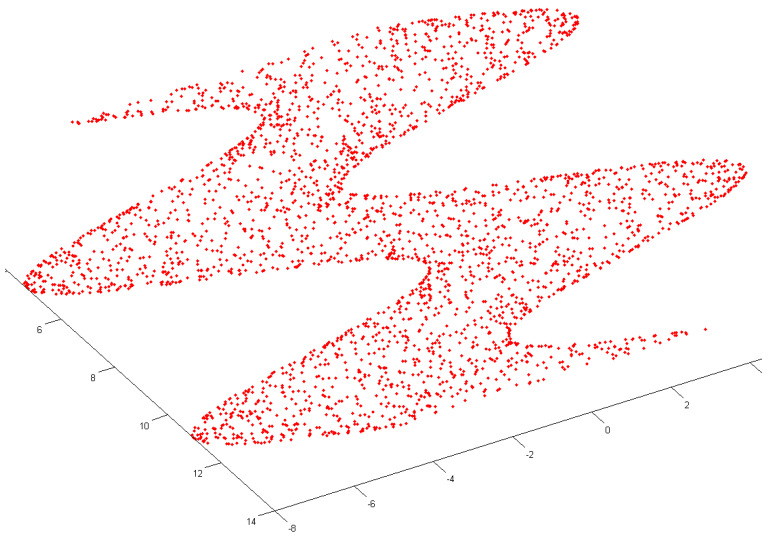


(a) Surface of Randomly Sampled U-Pipe



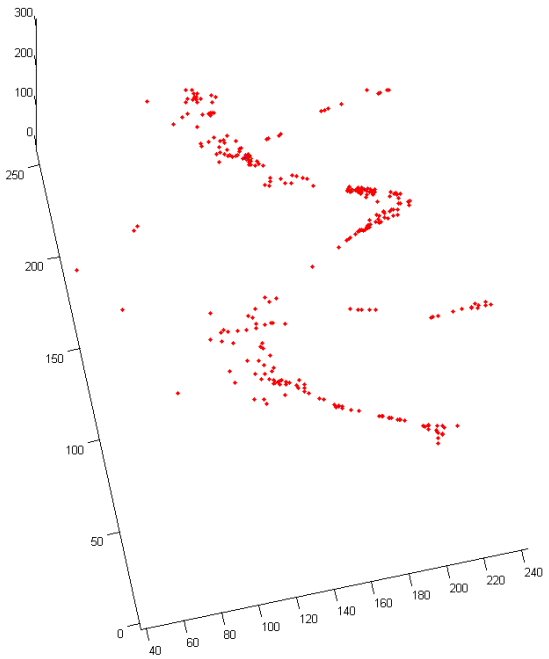
(b) Medial Axis of U-Pipe

Fig. 1.1: Shape and Estimated Medial Axis



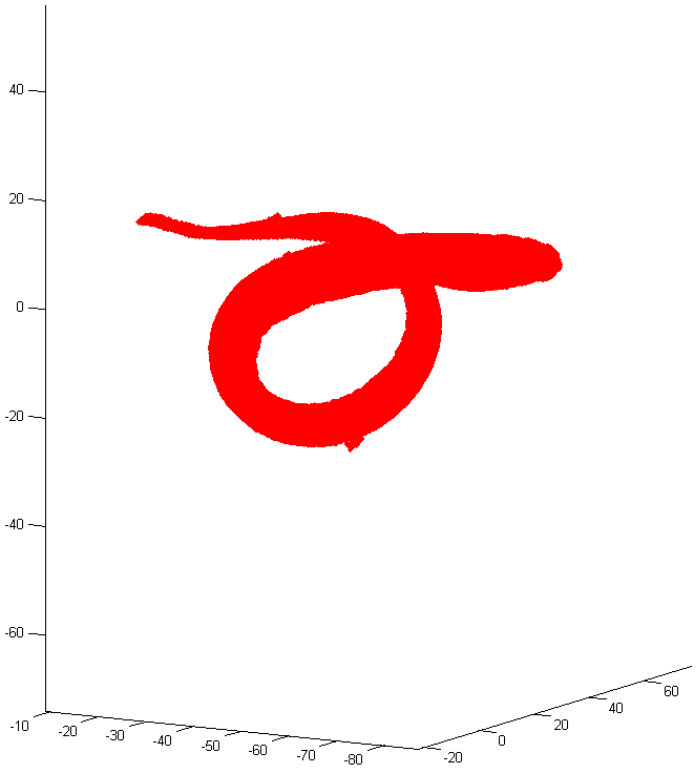
(a) Surface of Randomly Sampled Corkscrew

$t = 0,\ \sigma = 0.5,\ low = 0.3,\ high = 0.42$

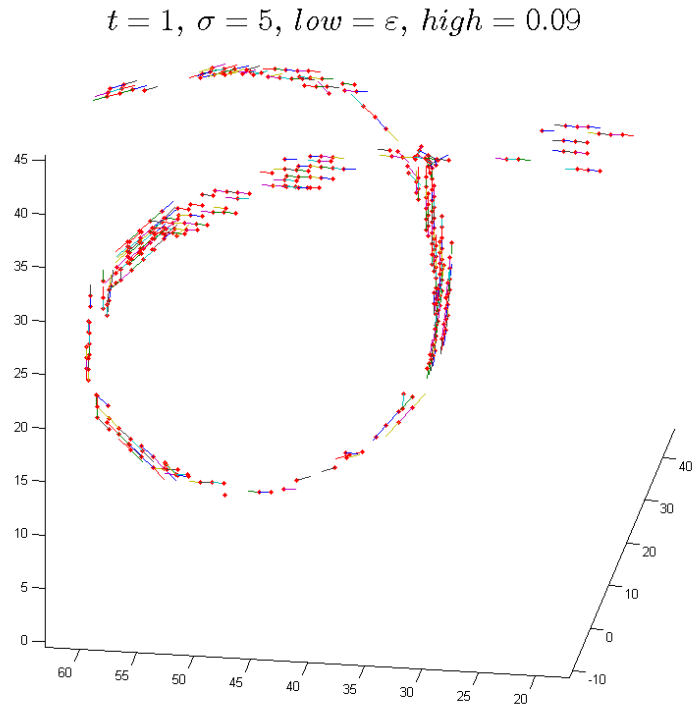


(b) Medial Axis of Corkscrew

Fig. 1.2: Shape and Estimated Medial Axis



(a) Surface of Randomly Sampled Eel Surface



(b) Medial Axis of Eel

Fig. 1.3: Shape and Estimated Medial Axis