A Voronoi based Labeling Approach to Curve Reconstruction and Medial Axis Approximation

Jiju Peethambaran, Amal Dev Parakkat, Ramanathan Muthuganapathy

Department of Engineering Design, Indian Institute of Technology Madras, Chennai - 600036, India

Abstract

In this paper, we present a Voronoi based algorithm for closed curve reconstruction and medial axis approximation from planar points. In principle, the algorithm estimates one of the poles (farthest Voronoi vertices of a Voronoi cell) and hence the normals at each sample point by drawing an analogy between a residential water distribution system and Voronoi diagram of input samples. The algorithm then labels Voronoi vertices as either inner *or* outer *with respect to the original curve and subsequently construct a piece-wise linear approximation to the boundary and the interior medial axis of the original curve for a class of curves having bi-tangent neighborhood convergence (BNC). The proposed algorithm has been evaluated for its usefulness using various test data. Results indicate that, even sparsely and non-uniformly sampled curves with sharp corners, outliers or collection of curves are faithfully reconstructed by the proposed algorithm.*

1. Introduction

Given a finite set of points $S \subseteq \mathbb{R}^2$, sampled from a smooth curve $\sum$, the task of constructing a polygonal chain from S faithful to $\sum$ is referred to as *curve reconstruction* problem. It is a fundamental yet challenging problem in areas such as computer graphics, computer vision, computational geometry and reverse engineering [Lee00, Wan14]. Curve reconstruction from an arbitrary data, insufficiently sampled from an unknown original curve, is almost infeasible [ABE98]. Hence, a few conditions on the sampling are needed to guarantee a faithful reconstruction of the original curve. Under uniform sampling, where the adjacent points are sampled at a distance less than a threshold value, many algorithms such as α-shape [EKS83] and r-regular shapes [Att98] are known to work with reasonable accuracy. However, uniform sampling condition leads to dense sampling all over the curve, including the areas where a sparse sampling is sufficient and hence represents a restrictive case of sampling to evaluate reconstruction algorithms.

To capture the local level of details, Amenta, Bern and Eppstein [ABE98] introduced a non-uniform sampling model called ε-sampling, where the sampling density varies with the *local feature size* on the curve. Under ε-sampling model, crust [ABE98] and its variants such as nearest neighbor crust [DK99] and a locally defined crust [Gol99] guarantee to construct a piece-wise linear approximation to Σ, for certain value of ε. Later, conservative crust [DMR00], that reconstructs a collection of open and closed smooth curves was described. It also showed better resistance towards noises and outliers at the expense of a parameter tuning.

Crust and its variants found to fail, both in theory and practice, for curves with sharp corners [DW02]. Aimed at curing this limitation, Dey and Wenger [DW01] described a heuristic called gathan that handles corners and endpoints in practice and subsequently, in [DW02], they extended 'gathan' to reconstruct a collection of piece-wise smooth closed curves with provable guarantee. In [FR01], Funke and Ramos proposed an algorithm based on empty β-balls to handle a collection of curves with corners and end points. A few notable work on curve reconstruction based on travelling salesman problem (TSP) can be found in [Gie99, AM01].

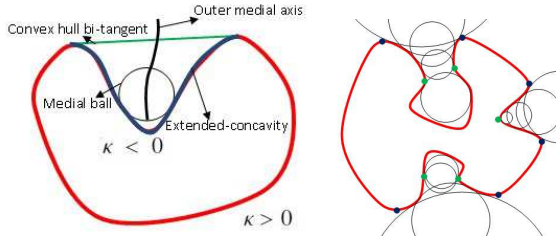
Most of the Delaunay-based approaches are parameter-based, which is not straight forward to identify for an optimal shape. Voronoi-based ones are application specific such as handling sharp corners, smooth curves, outliers etc. In this paper, the approach is based on a water distribution model (Section 3.1) that is non-parametric and not feature-specific.

2. Preliminaries

Let $\sum$ be a smooth closed and simple curve (1-manifold) embedded in $\mathbb{R}^2$. Let S be a set of n points sampled from $\sum$ and $Conv(\sum)$ denotes the convex hull of $\sum$. Further, $d(p,q) = \|p-q\|$, denotes the Euclidean distance between two points $p, q \in S$.

DEFINITION 1 Voronoi cell (V_p):
A Voronoi cell of $p \in S$ is the set of all points in the plane that are closer or at least equidistant to p than any other point in S: V_p=$\{x \in \mathbb{R}^2 \mid d(p,x) \leq d(q,x)$, where $p \neq q, \forall q \in S\}$.

Voronoi diagram (VD) of S, denoted by $Vor(S)$ is the sub division of the plane into Voronoi cells with one cell V_p for each point $p \in S$. $Vor(S)$ consists of **Voronoi bisectors also called as Voronoi edges (VE)**, as well as **Voronoi vertices (VV)**. A cell V_p is unbounded if the sample p lies on the convex hull of S. Unbounded Voronoi cells induce what is called as infinite edges, whose one vertex lies at infinity. All Voronoi vertices except the vertex at infinity are **finite**. The straight line dual graph of $Vor(S)$ results in a planar triangulation called as Delaunay triangulation of S, $Del(S)$.



(a) Extended-concavity (b) Smooth and BNC

Figure 1: *Illustration of different types of concavities of closed simple curves in 2D. In Figure 1(b), extended-concave portions between blue and green points represent the bi-tangent neighborhood convergent (BNC) portions.*

A *medial ball* $B(c,r)$, centered at $c \in$ medial axis (MA) of $\sum$ with radius r, is a maximal ball whose interior contains no points of $\sum$ [ABE98]. Let E be the set of all open connected regions of $Conv(\sum) \setminus \sum$. Each region given by the closure $\overline{E}$, is defined as an *extended concave region* ($\mathcal{C}_R$) of $\sum$. The portion of $\sum$ in each $\mathcal{C}_R$ is called *extended concavity*, denoted by $\mathcal{C}$ (Figure 1(a)). The edges of $Conv(\sum)$ in each *extended concave region* are called convex hull bi-tangents (BT_{cvx}). Every extended-concavity is capped by exactly one convex hull bi-tangent.

The medial balls of an extended-concavity, $\mathcal{C}$ tend to increase or decrease as it traverses through the outer MA lying in $\mathcal{C}_\mathcal{R}$. The region in $\mathcal{C}_\mathcal{R}$, where the medial ball monotonically increases or decreases is defined as a *rolling interval* of the medial ball.

DEFINITION 2 Bi-tangent neighborhood convergence (BNC):
Bi-tangent neighborhoods of an extended-concavity, $\mathcal{C}$ is said to be convergent, if the radius of the medial ball decreases monotonically in the first rolling interval, as it rolls along the outer medial axis of $\mathcal{C}$ from the convex hull bi-tangent end to its interior.

A curve is said to be bi-tangent neighborhood convergent, if all its extended-concavities are bi-tangent neighborhood convergent (Figure 1(b)). This characterisation of curves can be considered to be a generalisation of [PM15], where only closed curves called divergent concavity have been considered. Moreover. the algorithm based on water-flow and Voronoi diagram approach proposed in this paper is substantially different from the sculpting approach in [PM15].

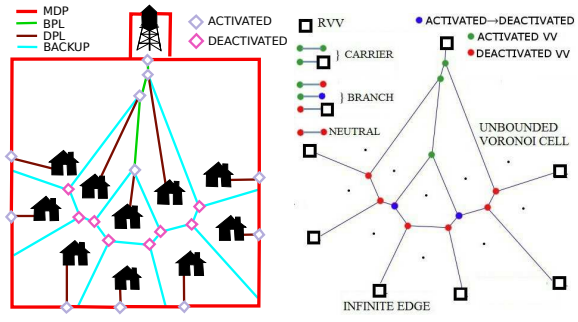
Like in [ABE98], we restrict our attention to sufficiently smooth curves which are twice differentiable and under this assumption, we can establish that all smooth, concave and closed planar curves are bi-tangent neighborhood convergent. Also, Using ε-sampling model by Amenta et al. [ABE98], it can be shown that $Vor(S)$, where S is densely sampled (ε-sample) from a BNC concave and closed planar curve $\sum$, has at least one finite Voronoi vertex outside $Conv(S)$.

3. Algorithm

In this section, we introduce the concept of water flow based curve reconstruction and MA extraction. Similar to other curve reconstruction methods [ABE98, Gol99, DMR00, DK99], we call our water distribution model based algorithm as WDM_CRUST and the corresponding interior MA as WDM_MAT.

3.1. Water Distribution Model (WDM) [WHM03]

Consider a water distribution system (a modified version of [WHM03]) in a residential area as shown in the Figure 2. It consists of a water tank erected at a considerable altitude, the main distribution pipes (MDP) which carry water to the residential area (red lines in the Figure 2(a)) and the branch pipe lines (BPL) that carry water to the different parts of the area (green lines in the Figure 2(a)). The houses get water either from MDP or BPL through the dedicated pipe lines (DPL), the dark brown lines in Figure 2(a). There exist few back up pipe lines (cyan lines in Figure 2(a)) which get activated only when the main distribution network fails.



(a) (b)

Figure 2: Water distribution model: *(a) water distribution system (b) corresponding VD illustrating different Voronoi vertices (VV) and the VE.*

To draw an analogy between the VD and the water distribution system, we consider each point in S, a house and each cell of $Vor(S)$, the land owned by the corresponding householder. The Voronoi edges represent the water distribution lines and each Voronoi vertex is a branching point of WDM. Further, we assume that the infinite edges of each unbounded Voronoi cell meet at an infinite point called root Voronoi vertex (RVV) (black squares in Figure 2(b)). RVV represents the water tank in WDM through which sufficient water may be pumped into the adjacent water distribution lines. It has openings to all the unbounded Voronoi cells. Similarly, each branching Voronoi vertex has openings to its adjacent Voronoi cells.

The finite Voronoi vertex of an infinite VE is referred to as farthest VV (FVV). If a farthest VV lies inside $Conv(S)$ (intuitively, $conv(S)$ represents the outer boundary of the residential colony), then it gets deactivated (red points in Figure 2(b)) due to the following reasons. First, most of these vertices lie inside the residential area and therefore, do not play a role in bringing water from the external sources. Second, the houses/points in the unbounded cells directly receive water through the corresponding DPL (refer to Figure 2(a)). Further, these act as back up lines and gets activated only when the main and branch lines fail. All other Voronoi vertices are activated at the start of water pumping (illustrated using green and blue points in Figure 2(b)).

3.2. State Transition of a Voronoi Vertex

Few Voronoi vertices (blue points in Figure 2(b)) are activated at the beginning of the water pumping and subsequently changed to the deactivated state during the process. The condition that triggers such a transition is based on the normal estimation technique proposed in [AB98]. Amenta and Bern [AB98] observed that Voronoi cells of S, where S is ε-sampled from a curve, $\sum$, tend to elongate in the direction of normal at each point. They define poles, which are two farthest Voronoi vertices of V_p, for each sample point p. In the case of curves, why the line passing through p and any of the two poles estimates the normal at p is explained in [DW01].

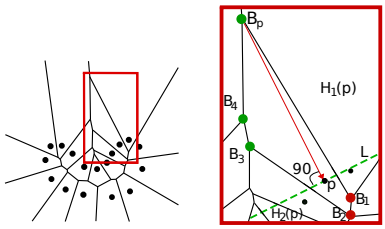


Figure 3: *A bounded Voronoi cell with the dedicated line (red line with arrow) and its perpendicular (green colored dashed line), activated (green points), Source VV (B_p) and deactivated (red points) VVs.*

In our model, each farthest Voronoi vertex (FVV) has exactly two unbounded Voronoi cells due to the infinite edge and one bounded Voronoi cell adjacent to it. During a pumping session, water flows into the FVVs from RVV. Please recall that a few of the farthest Voronoi vertices which lie inside $Conv(S)$ are deactivated at the start and hence block the flow of water. The remaining FVVs pass water to the adjacent VEs and to the dedicated lines of their corresponding adjacent bounded Voronoi cells. For a Voronoi cell V_p of a sample point, the activated Voronoi vertex from which it receives water represents its source VV (SVV). Each sample p in S has its own Voronoi cell V_p and hence one of the vertices of V_p subjected to a few other conditions, is guaranteed to be the SVV of p.

Consider a bounded Voronoi cell V_p and its source Voronoi vertex (represented as B_p) along with its owner point p as shown in Figure 3. The water flow in V_p assumes a direction of the vector $\overrightarrow{B_p, p}$. The line L, orthogonal to $\overrightarrow{B_p, p}$ divides the plane into two half planes designated as $H_1(p)$ and $H_2(p)$.

The water flow starts from the root vertex which is placed at infinity and advances towards $\sum$. Hence, it is obvious that the water reaches the SVVs of V_p before it reaches other Voronoi vertices of V_p. Otherwise, it would not have been the SVV of V_p. Consequently, SVV (B_p) is one of the farthest Voronoi vertices of V_p (refer to Figure 3) and hence represents one of its poles. Therefore, $-\overrightarrow{B_p, p}$ and L estimate the normal and the tangent at p respectively. Taking this idea forward, we present the state transition rule for a Voronoi vertex in our model in Definition 3.

DEFINITION 3 State transition rule:
Let B_i be an activated Voronoi vertex of a Voronoi cell V_p of a sample point p and B_p be the source Voronoi vertex of V_p, B_i is deactivated if B_p and B_i lie on either side of L

All the Voronoi vertices of V_p beyond p when viewed from B_p gets deactivated. The justification is that the Voronoi vertices that lie beyond p tend to lie inside the original curve (correspondingly, interior to the residential colony) and hence can be deactivated.

3.3. Classification of Voronoi Vertices and Edges

Next stage in our approach is to classify the Voronoi vertices and edges based on their location and functionality. From Figure 2(b), we can observe that all the deactivated and activated Voronoi vertices lie interior and exterior to the original curve, respectively. This is an interesting outcome of the water flow based approach, which can essentially be used to classify the vertices as *inner* and *outer* with respect to the original curve. Based on the observation, we classify all the Voronoi vertices in the deactivated state into *inner* class and all the Voronoi vertices in activated state into *outer* class. A similar labelling approach that depends on a locally defined crust [Gol99], has been adopted in [GMP07], to compute

the MA of the union of inner Voronoi balls, where as our approach is based on the water flow model. Hence, in addition to the MA approximation, our method is also capable of reconstructing the boundary of the input sample.

VEs are classified as *carrier*, *branch* and *neutral* edges (see Figure 2(b)) according to their functionality. *Carrier* edges with ACTIVATED (*outer*) vertices as end points represent the branch pipe line of the water distribution model. The back up lines (cyan lines in Figure 2(a)) of the water distribution system is represented by two classes of Voronoi edges. One class consists of *neutral* edges having both its Voronoi vertices labelled as DEACTIVATED (*inner*) and the other class consists of *branch* edges with an ACTIVATED (*inner*) and a DEACTIVATED (*outer*) end points. Please note that root Voronoi vertex is an activated vertex and therefore, all the infinite edges which has RVV and a deactivated vertex as its end points are *branch edges*.

Algorithm 1: Water_Flow(B_p)

Input: Branch Voronoi vertex, B_p
Apply state transition rule (Definition 3) to B_p;
if *there are no adjacent ACTIVATED and UNVISITED vertices for* B_p **then**
  return;
end
else if *there is one adjacent ACTIVATED and UNVISITED vertex for* B_p **then**
  Let B_{pnew} be the ACTIVATED adjacent vertex of B_p;
  Water_Flow(B_{pnew});
end
else
  Let B_{pnew1} and B_{pnew2} be the ACTIVATED adjacent vertices of B_p;
  Water_Flow(B_{pnew1});
  Water_Flow(B_{pnew2});
end

3.4. WDM_CRUST Extraction

To efficiently extract WDM_CRUST and WDM_MAT, we transform the water flow model into an algorithm based on efficient data structures and optimized geometric predicates. The backbone of the WDM_CRUST() algorithm is a recursive procedure called, Water_Flow() presented in Algorithm 1 which deactivates the required Voronoi vertices when the water flows t hrough the branch pipe lines. Each branch pipe line starts with a farthest VV denoted by B_p and hence constitutes the input parameter to the Water_Flow() procedure.

We assume that no four points are co-circular and hence each finite Voronoi vertex has a degree of 3. Each Voronoi vertex structure is equipped with state and visited fields to keep track of the current state and visited status during the Water_Flow() process. Water_Flow() procedure starts by applying state transition rule (Definition 3) on B_p to deactivate the eligible vertices of the Voronoi cell to which B_p is a SVV. The procedure continues till there are no adjacent unvisited vertices which are in active state for B_p.

Algorithm 2: WDM_CRUST(S)

Input: Point set S
Output: wdm_crust of S
Construct $Vor(S)$ and its dual $Del(S)$;
Initialize all the vertices of $Vor(S)$ including the INFINITE vertex to ACTIVATED state;
Construct a global list L_F containing $FVVs$ from $Vor(S)$, sorted in the descending order of their circum radii of the dual Delaunay triangles;
Deactivate and delete from L_F, all $FVVs$ which lie inside $Conv(S)$;
while L_F *not empty* **do**
  B_p=First(L_F);
  Water_Flow(B_p);
end
Extract the graph, $WDM\_CRUST(S) = \{e \mid$ edge $e \in Del(S)$ and $Dual(e)$ is a *branch* edge$\}$;
Extract the graph, $WDM\_MAT(S) = \{e \mid$ edge $e \in Del(S)$ and $Dual(e)$ is a *neutral* edge$\}$;
return *WDM_CRUST(S) & WDM_MAT(S)*

A pseudo code for WDM_CRUST is provided in Algorithm 2, which uses Algorithm 1. Key ideas include sorting of FVVs are sorted in the descending order of the circumradii of the corresponding dual Delaunay triangles and all the FVV lying interior to the convex hull are deactivated and deleted from L_F to ensure that the convex hull edges from the convex portion of the curve are included. The algorithm also extracts MA from $Del(S)$ by employing Dual() function which gives the dual Delaunay edge of a VE. The algorithm WDM_CRUST() has been designed to address closed curves, in particular extended concave curves and also not applicable for open curves. The worst case time complexity of WDM_CRUST() can be shown to be $O(n \log n)$, incurred due to the computation of Voronoi diagram. Theoretically, the following lemma can be proven.

LEMMA 3.1 $WDM\_CRUST(S)$, where S is ε-sampled from a BNC concave and closed planar curve $\sum$, contains an edge between every pair of adjacent samples of $\sum$, for $\varepsilon < 0.4$

4. Results and Discussion

We implemented our algorithm in C++ using computational geometry algorithms library (CGAL). To evaluate the performance of our approach, we tested it on points sampled randomly from the contours of silhouettes from MPEG 7 CE Shape-1 Part B and aim@shape repositories.

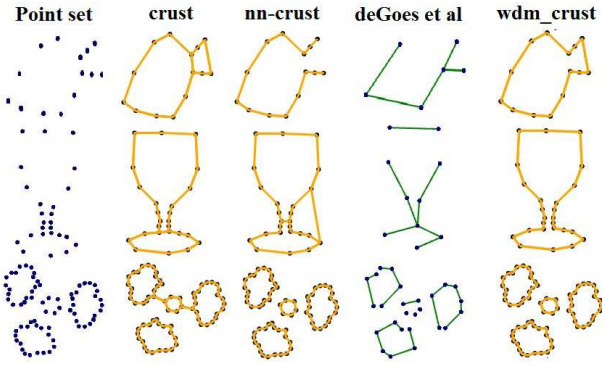


Figure 4: *Reconstruction from sparse data.* ***Rows (top to bottom)****: Point sets of fish, cup and collection of curves.* ***Columns (left to right)****: Point sets, crust [ABE98], nearest neighbor crust [DK99], result of [dGCSAD11] and WDM_CRUST*

Reconstruction from Sparse Data: Sparse data represents a major challenge for any curve reconstruction algorithm. In practice, *WDM_CRUST* algorithm is found to perform well for a variety of sparse and non-uniform input data as shown in Figure 4. For shapes such as fish and cup, *WDM_CRUST* is noticeably better than other crust algorithms and the optimal transport based algorithm [dGCSAD11]. Our method performs equally well in reconstructing a collection of closed curves from a sparsely sampled data as illustrated in the third row of Figure 4. Also compared to [PM15], our algorithm nicely reconstructs divergent as well as non-divergent concave portions of closed curves as shown in Figure 5.

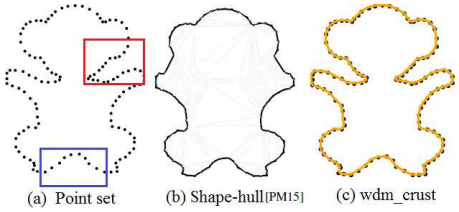


Figure 5: *(b) result from [PM15] works only for divergent concavity (blue box in (a)) (c) our approach works for non-divergent concavity as well.*

Robustness to Outliers: Most of the Delaunay/Voronoi based algorithms interpolate the input data and hence found to be intolerant towards the outliers. For point sets having noises and outliers, any curve fitting technique may be considered a more appropriate choice. Curve fitting techniques, however make implicit assumptions on the underlying curve, which is highly impractical for sparse and non-uniform data. Since WDM_CRUST is also an interpolating technique, rather than eliminating the outliers from the results, we aim at showing the reconstruction of the original shape while retaining outliers in the scene.

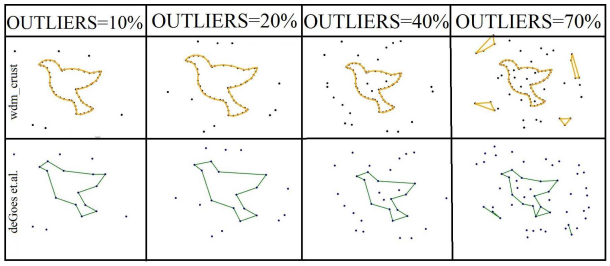


Figure 6: Outlier experiment: *All the stages of outlier injection, dove shape reconstructed by WDM_CRUST preserves fine details as compared to a simplified reconstruction by deGoes et al [dGCSAD11].*

We experimented on a dove point set consisting of 54 points. Random outliers, expressed as a percentage of dove point set size, were injected to the input data as shown in Figure 6. Our approach is noticeably better at dealing with the outliers constituting even 40% of the curve sample. Results by deGoes et al [dGCSAD11] lost many fine details of the dove shape even for 10% outliers. However, a few artifacts appear in the reconstruction for 70% outliers in the case of both the algorithms. Please note that, albeit these artifacts, dove shape has been well reconstructed by our method as opposed to the deGoes et al [dGCSAD11].

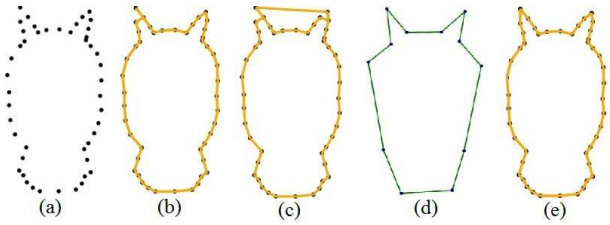


Figure 7: *Reconstruction of oni data. (a) Point set (b) crust [ABE98] (c) nearest neighbor crust [DK99] (d) result of [dGCSAD11] (e) WDM_CRUST.*

Dealing with Sharp Corners: On closed and BNC concave curves with sharp corners, our approach performs better than other methods. For instance, the left horn of oni which is sharp and pointed in Figure 7 is well captured by our algorithm as opposed to other crust algorithms. Though, optimal transport based approach reconstructs both the sharp corners well, it loses several other details such as neck of the oni. As opposed to this, our method not only captures the sharp corners but also preserves other details of the original curve. Figure 8 shows WDM_CRUST for point sets with sharp features. WDM_CRUST correctly reconstructs the shapes for

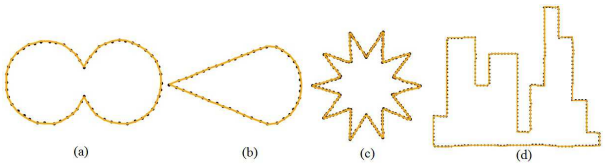


Figure 8: *WDM_CRUST of curves with sharp features.*

point sets in Figures 8 (a)-(b), for which all the TSP based algorithms listed in [AMNS00] fail.

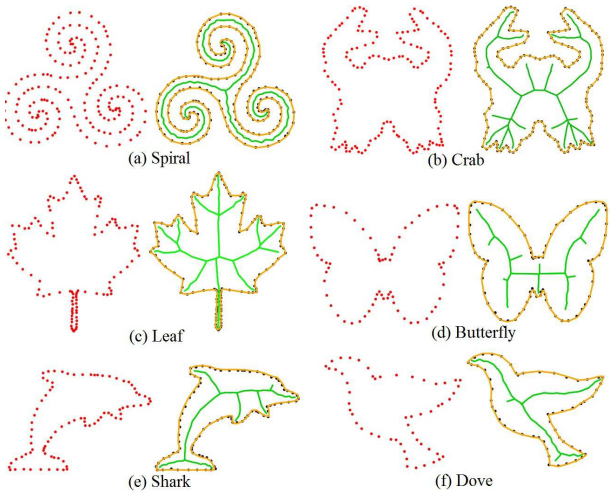


Figure 9: Medial Axes Gallery*: WDM_MAT generated for various non-uniformly sampled data.*

Medial Axis Results: Figure 9 shows the reconstructed curves as well as the medial axes for various non-uniformly sampled data. Like any other approach, the approximation quality of our medial axis algorithm is limited by the sampling density of input data and apparently the smoothness of the given curve.

5. Conclusion

In this paper, we presented a Voronoi and Delaunay based algorithm for curve reconstruction and medial axis approximation using the idea of water flow model and for closed and concave planar curves that are bi-tangent neighborhood convergence. Experimental results indicate that our approach is capable of capturing sharp corners and reconstructs the curves from sparse data. Currently, extension of the proposed algorithm to three dimensions is being considered.

References

[AB98] AMENTA N., BERN M.: Surface reconstruction by voronoi filtering. In *Proceedings of the Fourteenth Annual Symposium on Computational Geometry* (New York, NY, USA, 1998), SCG '98, ACM, pp. 39–48.

[ABE98] AMENTA N., BERN M., EPPSTEIN D.: The crust and the beta-skeleton: Combinatorial curve reconstruction. In *Graphical Models and Image Processing* (1998), pp. 125–135.

[AM01] ALTHAUS E., MEHLHORN K.: Traveling salesman-based curve reconstruction in polynomial time. *SIAM Journal on Computing 31*, 1 (2001), 27–66.

[AMNS00] ALTHAUS E., MEHLOHRN K., NAHER S., SCHIRRA S.: Experiments on curve reconstruction. In *Proceedings of 2nd Workshop on Algorithms Enginering experiments* (2000), pp. 104–114.

[Att98] ATTALI D.: r-regular shape reconstruction from unorganized points. *Computational Geometry 10*, 4 (1998), 239 – 247.

[dGCSAD11] DE GOES F., COHEN-STEINER D., ALLIEZ P., DESBRUN M.: An optimal transport approach to robust reconstruction and simplification of 2d shapes. *Computer Graphics Forum 30*, 5 (2011), 1593–1602.

[DK99] DEY T. K., KUMAR P.: A simple provable algorithm for curve reconstruction. In *Proceedings of the Tenth Annual ACM-SIAM Symposium on Discrete Algorithms* (1999), SODA '99, pp. 893–894.

[DMR00] DEY T. K., MEHLHORN K., RAMOS E. A.: Curve reconstruction: Connecting dots with good reason. *Computational Geometry 15*, 4 (2000), 229 – 244.

[DW01] DEY T. K., WENGER R.: Reconstructing curves with sharp corners. *Computational Geometry 19*, 2-3 (2001), 89 – 99.

[DW02] DEY T. K., WENGER R.: Fast reconstruction of curves with sharp corners. *International Journal of Computational Geometry & Applications 12*, 05 (2002), 353–400.

[EKS83] EDELSBRUNNER H., KIRKPATRICK D., SEIDEL R.: On the shape of a set of points in the plane. *Information Theory, IEEE Transactions on 29*, 4 (jul 1983), 551 – 559.

[FR01] FUNKE S., RAMOS E. A.: Reconstructing a collection of curves with corners and endpoints. In *Proceedings of the Twelfth Annual ACM-SIAM Symposium on Discrete Algorithms* (2001), SODA '01, pp. 344–353.

[Gie99] GIESEN J.: Curve reconstruction, the traveling salesman problem and menger's theorem on length. In *Proceedings of the Fifteenth Annual Symposium on Computational Geometry* (New York, NY, USA, 1999), SCG '99, ACM, pp. 207–216.

[GMP07] GIESEN J., MIKLOS B., PAULY M.: Medial axis approximation of planar shapes from union of balls: A simpler and more robust algorithm. In *Proceedings of the 19th Annual Canadian Conference on Computational Geometry, CCCG 2007, August 20-22, 2007, Carleton University, Ottawa, Canada* (2007), pp. 105–108.

[Gol99] GOLD C.: Crust and anti-crust: A one-step boundary and skeleton extraction algorithm. In *Proceedings of the Fifteenth Annual Symposium on Computational Geometry* (New York, NY, USA, 1999), SCG '99, ACM, pp. 189–196.

[Lee00] LEE I.-K.: Curve reconstruction from unorganized points. *Computer Aided Geometric Design 17*, 2 (2000), 161 – 177.

[PM15] PEETHAMBARAN J., MUTHUGANAPATHY R.: Reconstruction of water-tight surfaces through delaunay sculpting. *Computer-Aided Design 58*, 0 (2015), 62 – 72.

[Wan14] Robust reconstruction of 2d curves from scattered noisy point data. *Computer-Aided Design 50*, 0 (2014), 27 – 40.

[WHM03] WALSKI T., HAESTAD METHODS I.: *Advanced water distribution modeling and management*. Haestead Press, 2003.