An RNG-based heuristic for curve reconstruction

Asish Mukhopadhyay
School of Computer Science
University of Windsor, Canada
e-mail: asishm@cs.uwindsor.ca *

Augustus Das

Abstract

In this paper we propose an RNG-based heuristic for curve reconstruction. It is extremely simple and works remarkably well in practice. To establish its superiority more conclusively, we include comparisons of the outputs of our algorithm with established algorithms reported in [7] and [6] for a large variety of samples.

1 Introduction

The curve reconstruction problem is to reconstruct an unknown curve $\mathcal{C}$ in a given class (for example smooth and closed, or smooth and open, regular etc.) from a sample S of n points $\{p_1, p_2, p_3, \ldots, p_n\}$ (See Fig. 1). In view of the practical importance of the problem, in GIS for example (see [8], [9]), it has received quite a bit of attention from researchers. The main thrust of some of the recent work has been on reconstructing curves whose correctness are guaranteed provided the sample S, assumed drawn from a curve of a specified class, satisfies a sampling condition. We will briefly review some of this work in the next section.

The work that we report in this paper follows this line of work. We propose a curve reconstruction algorithm that is based on a subgraph of the Delaunay triangulation, known as the Relative Neighborhood Graph, RNG for short, proposed more than two decades ago by [13]. It work extremely well on a variety of samples, that simulate smooth single curves, nested curves, multiple curves as well as curves that other researchers have used to support the effectiveness of their reconstruction. To support this claim we have provided comparisons of the outputs of our algorithm with those of two widely referenced algorithms published in [7] and [6].

The rest of the paper is organized as follows. In the next section we briefly review previous related work on this problem. In the following section we give a very brief discussion of the RNG and other terminolgy. In the fourth section we describe our heuristic, along with an anlysis of its running time. We also briefly discuss another heuristic. In the fifth and final section, we conclude. The output of our heuristic on a variety of samples as well those of [7] and [6] are provided in an appendix.

2 Previous Work

Many of the keys ideas underpinning some of the recent Delaunay triangulation-based reconstruction algorithms originate in the work by Brandt and Algazi [5], who showed how to obtain the skeleton of an r-regular shape from the Voronoi diagram of a set of points sampled along the boundary of that shape.

*A preliminary version of this paper appeared in ISVD 2006, Banff, Alberta, Canada, [12]

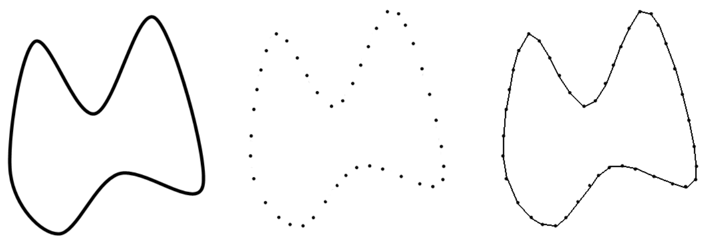


Figure 1: The original Curve, Sample Points and the reconstructed Curve

The parameter r controls two aspects of such a shape: the curvature at a boundary point never exceeds the reciprocal of r, and the radius of a maximal disk contained in this shape or its complement never exceeds r. They showed that for shapes in this class the computed skeleton converges to the exact skeleton as the sampling density increases.

Dominique Attali [4] borrowed this notion of an r-regular shape and provided a correct reconstruction for shapes in this class under some guarantee on the sample. Her ideas were further refined and amplified by subsequent researchers. Amenta, Bern and Epstein [3] showed how to reconstruct a smooth curve from a non-uniform ϵ-sample. Their most important contribution was the idea of local feature size $f(p)$ at a point p on the unknown curve $\mathcal{C}$, which determines the sampling density in the neighbourhood of p. Subsequently Dey and Kumar [6] considerably simplified the ideas of [3] to correctly reconstruct smooth closed curves in any dimension from an ϵ-sample, where $\epsilon \leq 1/3$. In a following paper [7] Dey et al proposed a rather complex algorithm for the polygon reconstruction from a sample, not necessarily from a curve that is smooth and closed, whose correctness is witnessed by a smooth curve that they construct. In other words they "invent" a curve whose correct reconstruction is the output of their algorithm. All of the above algorithms use the Delaunay triangulation to look for short edges locally around sample points.

In a significant departure, Giesen [8] proposed a global minimization technique based on computing a Travelling Salesman Tour of the sample points for curves which satisfy a regularity assumption. Subsequently, Althaus and Melhorn [2] showed how to compute such a tour in polynomial time. Guha and Tran [10] also proposed a non-Delaunay-based technique that reconstructs a curve in 2 or 3 dimensions from a sample of points S. Their proposed method determines monotone pieces of the curve, using the idea of bounding curvature.

3 Some terminology

The *Relative Neighbourhood Graph* (RNG) was invented by Toussaint [13]. For a given point set S, an edge joining two points p_i and p_j is in the RNG if the lune defined by the edge $\overline{p_i p_j}$ does not contain any other point of S. The lune is the region of intersection of the circles centred at p_i and p_j, each with radius equal to $length(\overline{p_i p_j})$. The RNG on a sample set of points is shown in Fig. 2. It can be shown that the RNG is a subgraph of the Delaunay triangulation on S [13].

The *Gabriel Graph* on S consists of the set of edges joining pairs of points, such that the diametral disk on each such edge contains no other points from S. This is also provably a subgraph of the Delaunay triangulation on S [11].

The *medial axis* of a smooth curve $\mathcal{C}$ is the locus of the centers of maximal disks that touch the curve at two or more points.

The *local feature size* $f(p)$ at a point p on the curve $\mathcal{C}$ is the distance from p to the nearest point on the medial axis.

An ϵ-sample S satisfies the following condition: for any point p on the curve $\mathcal{C}$, the distance to the nearest sample point is at most $\epsilon * f(p)$.

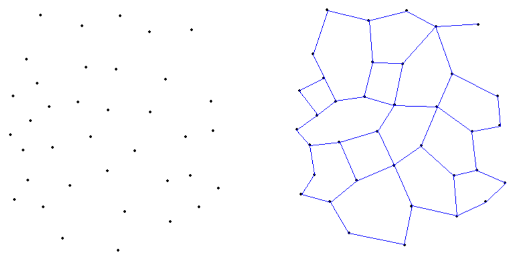


Figure 2: A point set and its RNG

4 RNG-Algorithm

Our work was triggered by the result in [7], where the starting point is the Gabriel graph and our motive was to determine if it would not be more advantageous to start with a sparser graph, such as the Relative Neighbourhood Graph (RNG). As our experimental results in a later section show, this is indeed the case.

Let H denote the graph on S that has edges only between pairs of points that are adjacent on $\mathcal{C}$.

We show below that the RNG contains all the edges of H, provided S is an ϵ-sample with $\epsilon \leq 1/5$. To establish this, we use the following lemmas.

Lemma 1 *Let S be an ϵ-sample from a smooth, planar curve $\mathcal{C}$ and H be its polygonal reconstruction from S. Then*

1. *If a closed disk B contains 2 or more points $\mathcal{C}$, then its intersection of with B is either a toplogical 1-disk or contains a point of the medial axis [3].*
2. *For $\epsilon < 1$, if ab is an edge of H, then $length(ab) < 2\epsilon/(1-\epsilon) * f(p)$, where $f(p)$ is the local feature size at p [6].*

While we omit the proofs that can be found in the papers cited, a few comments are pertinent. The assumptions about the nature of the curve that the sample comes from is used in the proof of part(1) above. The result in the second part is very interesting. We can use this to eliminate long edges from any supergraph of H. The right hand side shows a beautiful functional dependance of the length of an edge in H with the sampling density and the curvature (local feature size) of the unknown curve $\mathcal{C}$ at a point p.

Lemma 2 *Any Voronoi disk (a maximal empty disk centered at a Voronoi vertex of a Voronoi diagram) of a set of sample points S of a smooth, planar curve, $\mathcal{C}$, must contain a point of the medial axis of $\mathcal{C}$. [3]*

Our main result is contained in the following theorem.

Theorem 1 *If S is an ϵ-sample from a smooth closed curve $\mathcal{C}$, where $\epsilon \leq 1/5$, the graph H is a subgraph of the RNG on S.*

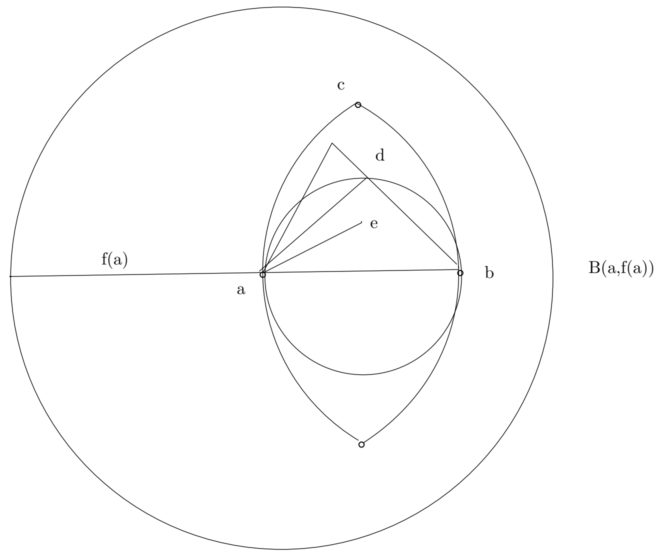


Figure 3: An edge joining adjacent points along the curve is in the RNG

Proof: Let $\overline{ab}$ be an edge joining two curve-adjacent points a and b in S. (Fig. 3). Let c be a point of S that lies inside the lune of $\overline{ab}$. We consider two cases that can arise.

- c is also inside the diametral disk of $\overline{ab}$.

  We prove the impossibility of this case by using a similar argument as in [7]. Since $\epsilon \leq 1/5$ it is, a fortiori, less than 1 and hence $f(a) > length(\overline{ab}) * (1-\epsilon)/2\epsilon$ by Lemma 1, part(2). This means that the disk $B(a, f(a))$ contains the disk with ab as diameter. If c were inside this disk then it would have to intersect $\mathcal{C}$ in more than one component, and hence would have to contain a medial axis point by Lemma 1, part(1). This would imply that $B(a, f(a))$ would have to contain a medial axis point, contradicting the definition of the feature point function, $f()$.

- c is outside the diametral circle of $\overline{ab}$.

  The triangle $\triangle abc$ is acute-angled since $\angle adb = 90$ degrees $> \angle acb$ (Fig. 3), while ac and bc being inside the lune, they cannot make an angle > 90 degrees with ab. Thus the center, e, of the circumcircle of $\triangle abc$ lies inside the triangle in question. Therefore, $2 * length(\overline{ae})$, the diameter of the circumcircle of $\triangle abc$ is $< 2 * length(\overline{ab})$. The last inequality is true because the point e is inside the lune of ab and hence inside the circle whose center is at a and has radius $= length(\overline{ab})$.

  But then again by Lemma 1, part (2), $f(a) > (1-\epsilon)/2\epsilon * length(\overline{ab}) > (1-\epsilon)/2\epsilon * length(\overline{ae})$. We can therefore ensure that the circumcircle of $\triangle abc$ is inside $B(a, f(a))$ by letting $(1-\epsilon)/2\epsilon > 2$ or $\epsilon < 1/5$. We can now repeat the argument for the circumcircle of $\triangle abc$ as we did for the diametral disk of ab for the first case if it contains some other sample point; or if it is point-free it must be a Voronoi disk and hence contain a point of the medial axis by Lemma 2. $\square$

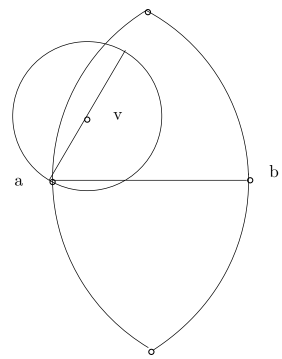
The most novel and significant contribution of this paper is the following simple heuristic for removing the edges that join pairs of non-adjacent points along the unknown curve C. It is as follows.

We construct the Voronoi diagram on S. For each RNG edge $\overline{p_i p_j}$, we compute the maximum distance from p_i to the vertices of its Voronoi polygon. Let this distance be d_i; do the same for the point p_j, obtaining a distance d_j. We delete the edge $\overline{p_i p_j}$ if its length is greater than the maximum of d_i and d_j.

We tried to eliminate RNG-edges that join pairs of points that are not curve-adjacent, using a different heuristic, based on the following theorem. The underlying intuition was that a long edge of this type would have to cross the medial axis.

Lemma 3 *If there is a Voronoi vertex inside the lune of an edge $\overline{ab}$ then the points a and b are not adjacent along the curve.*

Proof: Let there be a Voronoi vertex, v, inside the lune of an edge $\overline{ab}$ of length $l(\overline{ab})$, so the distance of a from v is at most $l(\overline{ab})$. Now, we know that a maximal Voronoi disk centered at v contains a medial axis point. Let D be a disk with v as center, passing through the closer of a or b. Either D is maximal or contains the maximal disk. Thus D contains a medial axis point. So the maximum distance of a or b from this medial axis point is at most $2l(\overline{ab})$. Thus $f(a)$, the feature point function at a, is at most $2l(\overline{ab})$.



We also know that if $\overline{ab}$ is an edge joining curve- adjacent points then

$$l(\overline{ab}) > \frac{2\epsilon}{1-\epsilon} f(a) \tag{1}$$

from which it follows that

$$f(a) > \frac{1-\epsilon}{2\epsilon} l(\overline{ab}) \tag{2}$$

Now since $0 < \epsilon \leq 1/5$ in our RNG-based reconstruction, $f(a) > 2l(\overline{ab})$, for all ϵ in this range, contradicting our conclusion from the above argument. $\square$.

However, this did not work as successfully as the first heuristic and so we abandoned it in favour of the first.

Here is a formal description of the algorithm.

CR Algorithm

Input: A set of sample points S from an unknown smooth curve C
Output: A poygonal reconstruction of C

Step 1. Compute the Delaunay Triangulation, DT on S.
Step 2. Extract the RNG from the DT.
Step 3. Compute the Voronoi Diagram, VD, as the dual of the DT obtained in Step 1.
Step 4. For each edge $\overline{p_i p_j}$ of the RNG computed in Step 2 do:

Step 4.1 Compute the maximum distance d_i from p_i to the vertices of its Voronoi polygon.
Step 4.2 Compute the maximum distance d_j from p_j to the vertices of its Voronoi polygon.
Step 4.3 Set $d_{max} = max(d_i, d_j)$.
Step 4.4 If $d_{max} < length(\overline{p_i p_j})$, delete edge $\overline{p_i p_j}$.

Step 5. Output the remaining set of edges.

The time-complexity of the RNG-algorithm is clearly in $O(n \log n)$ as the construction of the Delaunay triangulation in Step 1 dominates all other steps.

The algorithm has been implemented in Java and the program was run on a variety of sample points. Sample runs of the algorithm on collection of sample points drawn from different types of curves are shown in Figs. 4, 5.

As can be seen from the sample runs, our algorithm successfully reconstructs simple curves, curves with sharp corners, nested curves, and collection of curves. The algorithm is also able to handle to some degree curves with end points. However the density of sampling at the sharp corners of a curve have to be higher than for a normal smooth curve.

In an appendix we have compared the outputs of our algorithm with those reported in [7], [6]. Since we didn't have access to the source codes of these latter algorithms, we resorted to brute-force implementations. This didn't matter as our goal was only to compare the outputs.

5 Concluding remarks

Surely, we have not shown that the above heuristic removes all RNG-edges that join non-curve adjacent pairs of points. That's because it is not possible to do so if one starts with a supergraph of H.

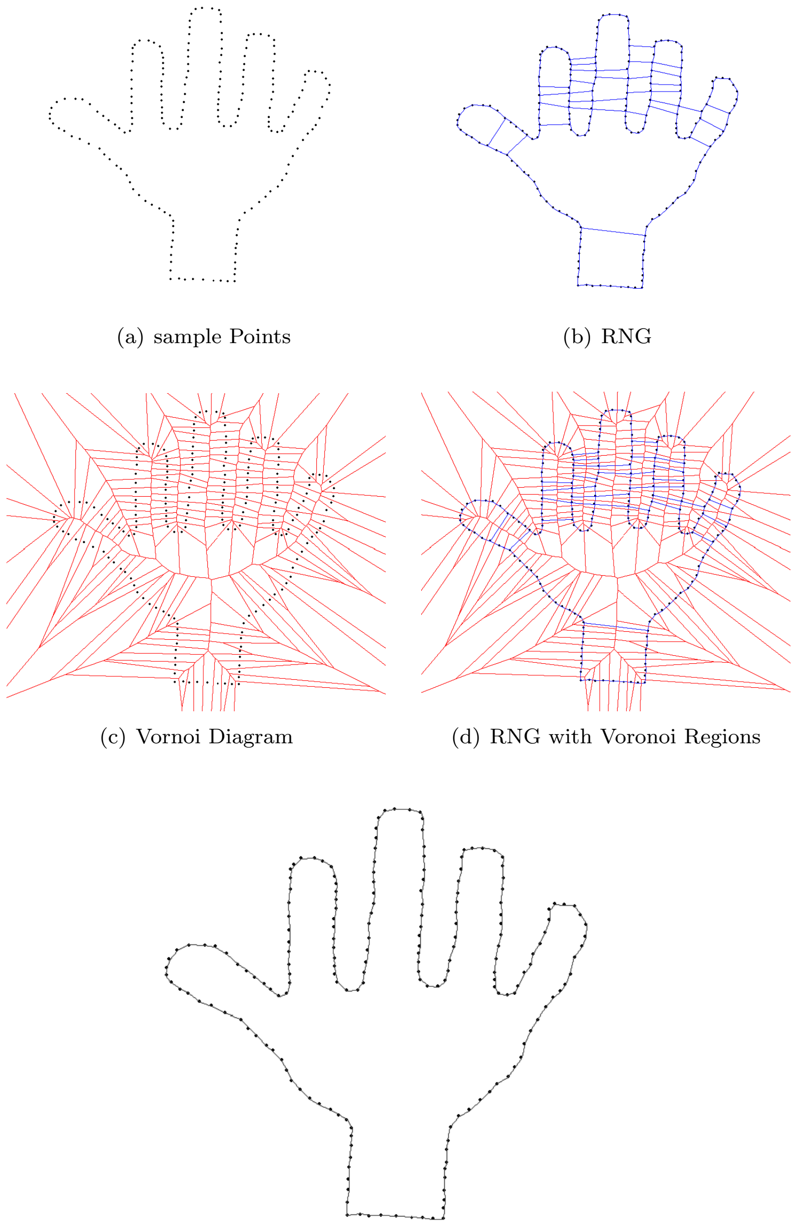
A close look shows that there are two kinds of edges that we should remove - long edges which cross the medial axis and long edges which do not. We have relied on the RNG-construction to exclude the latter type edges. The heuristic has been designed to remove the edges of the former type. It implicitly uses the condition in part(ii) of Lemma 1. For $\epsilon \leq 1/5$, $l(e) \geq 2/5/(4/5)f(p) \geq 2\epsilon/(1-\epsilon)f(p)$. Since we don't know $f(p)$, we replace this by measuring the distance from p to a Voronoi vertex of its Voronoi polygon that is farthest away. Thus we use the test $l(e) > max(f(p), f(q)), e = \overline{pq}$, where $f(p)$ and $f(q)$ are replaced by their estimates, to remove a long edge that crosses the medial axis. This is a stronger version of the inequality $l(e) > 2\epsilon/(1-\epsilon)f(p)$ and hence we might miss some edges that whose lengths lie between $max((f(p), f(q))$ and $2\epsilon/(1-\epsilon)$.

The same situation arises in [7]. As a sop they construct a smooth curve Γ that is said to witness the correctness of their reconstruction. More simply, the points of S are incident on this curve and the output of their algorithm consists exactly of edges joining the curve-adjacent points of S on Γ. In the literature this is referred to as the algorithm being self-consistent [1], meaning that it never "invents" a new curve when additional sample points are provided. We have not been able to make much sense out of this.

It is interesting to observe that the NN-crust algorithm that starts with a subgraph of H, and outputs a graph that is exactly H when $\epsilon \leq 1/3$. However, quite strangely this algorithm is the one that fares the worst in practice as can be seen from the oututs of the samples given in a later section. We don't seem to know how to explain this anamolous behaviour.

Though the theoretical guarantee of our algorithm ($\epsilon \leq 1/5$) is worse than that of the NN-crust algorithm, run on the same samples as shown in the appendix, our algorithm coinsistently outperforms the NN-crust algorithm.

Two interesting avenues stem from this work that seem worthy of exploration - one is to examine the application of this algorithm in GIS-related work; the other is to try to extend this algorithm to 3-dimensions. We plan to pursue these ideas with future publications.



(a) sample Points

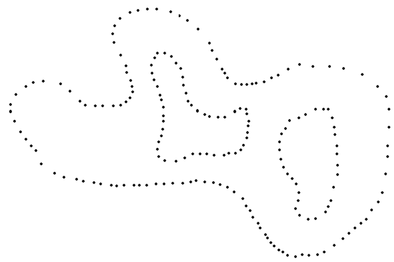
(b) RNG

(c) Vornoi Diagram

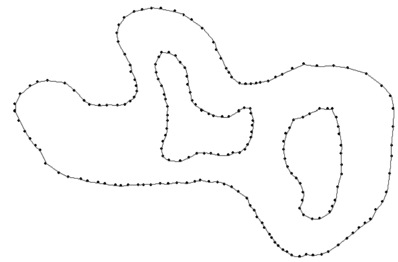
(d) RNG with Voronoi Regions

(e) Reconstructed Curve

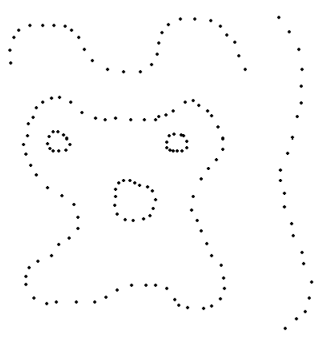
Figure 4: Steps of the Reconstruction Algorithm



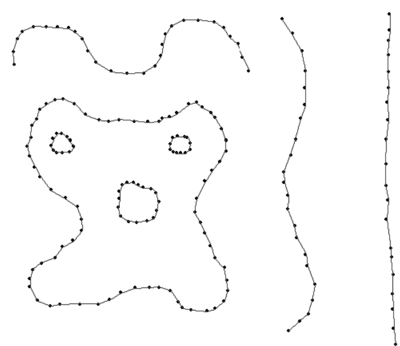
(a) sample Points



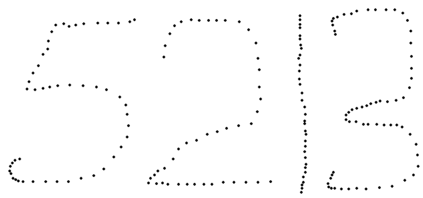
(b) Reconstructed Curve



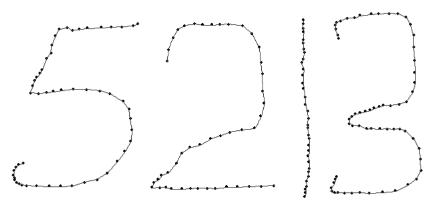
(c) sample Points



(d) Reconstructed Curve



(e) sample Points



(f) Reconstructed Curve

Figure 5: Reconstruction on three different sets of sample points using CR Algorithm

6 Acknowledgements

This research has been supported by an NSERC Discovery Grant to the first author. We also acknowledge the help of Bryan St. Amour and Chong Wang in producing the sample outputs.

The proof of Lemma 3 is due to Harshit Rathod of IIT Kharagpur.

References

[1] E. Althaus, K. Mehlhorn, S. N?her, and S. Schirra. Experiments on curve reconstruction. In *In Proc. 2nd Workshop Algorithm Eng. Exper*, pages 103–114, 2000.

[2] E. Althaus and K. Melhorn. Polynomial time tsp-based curve reconstruction. In *Proc. 11th ACM-SIAM Symposium on Discrete Algorithms*, pages 686–695, 2000.

[3] N. Amenta, M. Bern, and D. Eppstein. The crust and the β-skeleton: Combinatorial curve reconst ruction. In *Graphical Models and Image processing*, volume 60, pages 125–135, 1998.

[4] D. Attali. r-regular shape reconstruction from unorganized points. In *Proc. 13th Annu. ACM Sympos. Comput. Geom.*, pages 248–253, 1997.

[5] J. W. Brandt and V. R. Algazi. Continuous skeleton computation by Voronoi diagram. *CVGIP: Image Understanding*, 55(3):329–338, 1992.

[6] T. Dey and P. Kumar. A simple provable algorithm for curve reconstruction. In *ACM-SIAM Symposium Discr. Algorithms*, pages 893–894, 1999.

[7] T. Dey, K. Mehlhorn, and E. A. Ramos. Curve reconstruction : connecting dots with good reason. In *Proc. 15th Annual ACM Sympos. Comput. Geom.*, pages 197–206, 1999.

[8] C. M. Gold. Crust and anti-crust: a one step boundary and skeleton extraction algorithm. In *15th. ACM Symposium on Computational Geometry*, pages 189–196, 1999.

[9] C. M. Gold, D. Mioc, F. Anton, O. Sharma, and M. Dakowicz. A methodology for automated cartographic data input, drawing and editing using kinetic delaunay/voronoi diagrams. In M. L. Gavrilova, editor, *Generalized Voronoi Diagram*, volume 158 of *Studies in Computational Intelligence*, pages 159–196. Springer, 2008.

[10] S. Guha and S. D. Tran. Reconstructing curves without delaunay computation. *Algorithmica*, 42:75–94, 2005.

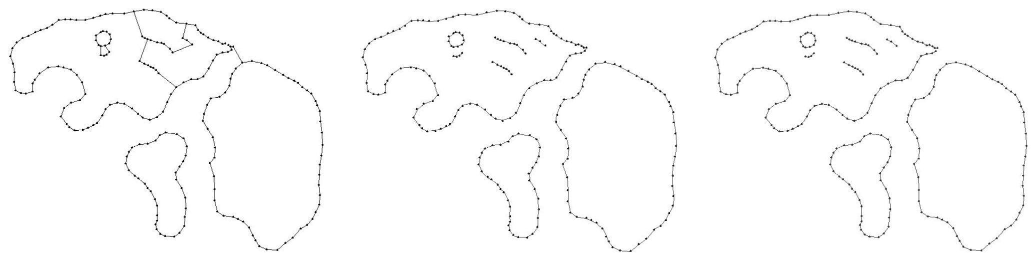
[11] D. W. Matula and R. Sokal. Properties of gabriel graphs relevant to geographic variation research and clustering of points in the plane. *Geogr. Anal.*, 12:205–222, 1980.

[12] A. Mukhopadhyay and A. Das. An rng-based heuristic for curve reconstruction. In *ISVD '06: Proceedings of the 3rd International Symposium on Voronoi Diagrams in Science and Engineering*, pages 246–251, Washington, DC, USA, 2006. IEEE Computer Society.

[13] G. T. Toussaint. The relative neighbourhood graph of a finite planar set. *Pattern Recogn.*, 12:261–268, 1980.

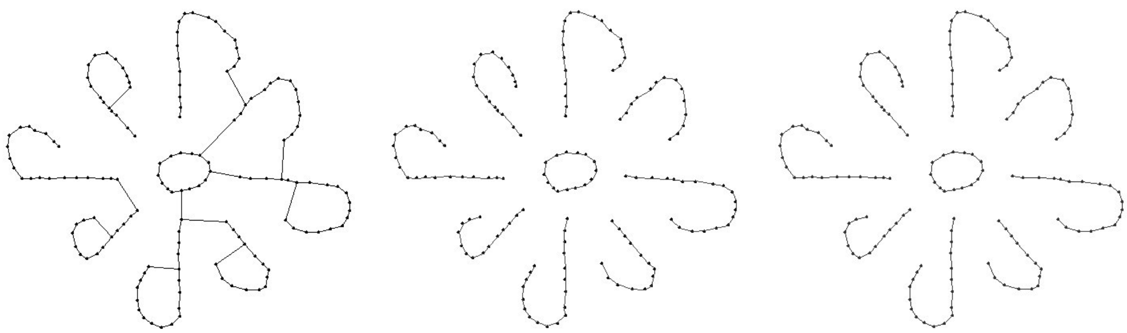
7 Appendix A

The figures below show the outputs of our algorithm and the algorithms of [7], [6] on a set of sammples that simulate an entire gamut of curves, ranging from smooth-and-closed to one that consists of a collection of curves. The output of the [7] paper had be fine-tuned to discover the parameter ρ that produces the best output.



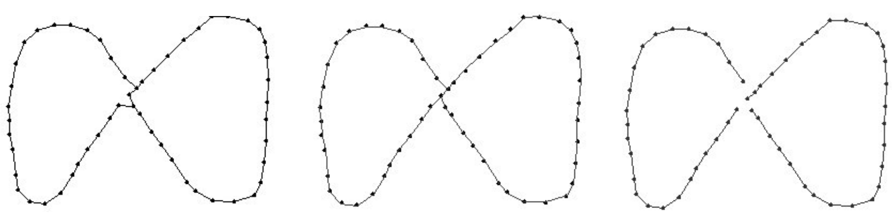
(a) NN-Crust (b) RNG (c) Conservative Crust

Figure 6: Sample 1



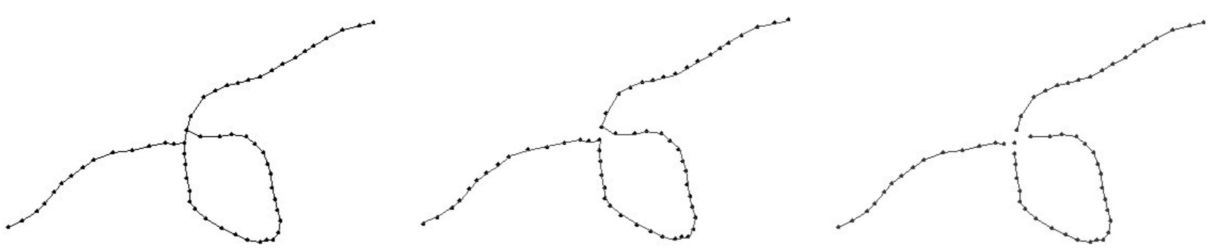
(a) NN-Crust (b) RNG (c) Conservative Crust

Figure 7: Sample 2



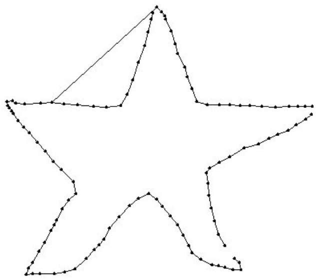
(a) NN-Crust (b) RNG (c) Conservative Crust

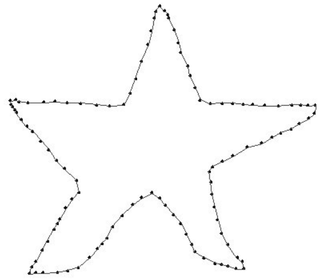
Figure 8: Sample 3



(a) NN-Crust (b) RNG (c) Conservative Crust

Figure 9: Sample 4


(a) NN-Crust


(b) RNG

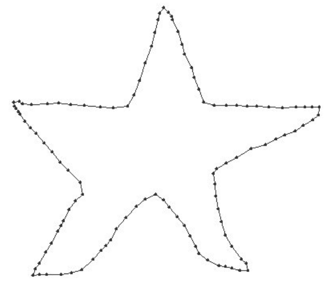
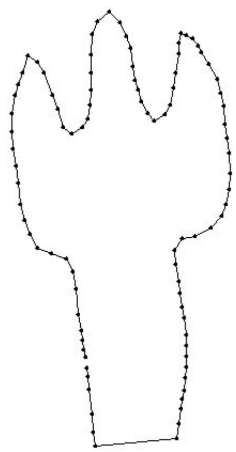
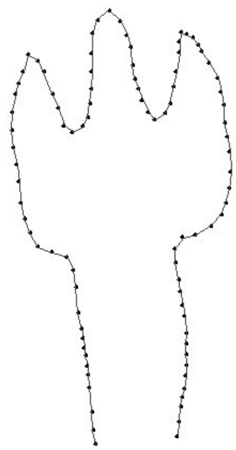

(c) Conservative Crust

Figure 10: Sample 5


(a) NN-Crust


(b) RNG

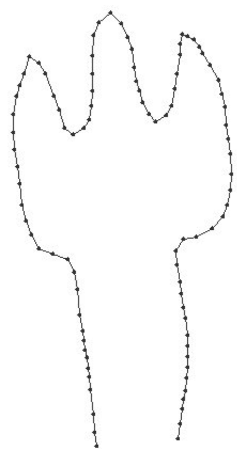
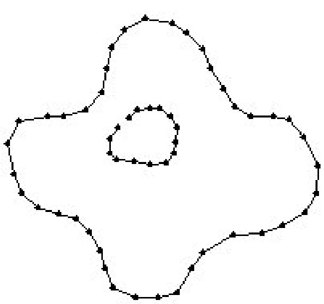
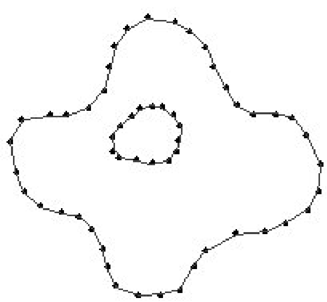

(c) Conservative Crust

Figure 11: Sample 6


(a) NN-Crust


(b) RNG

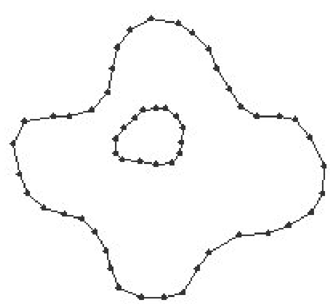
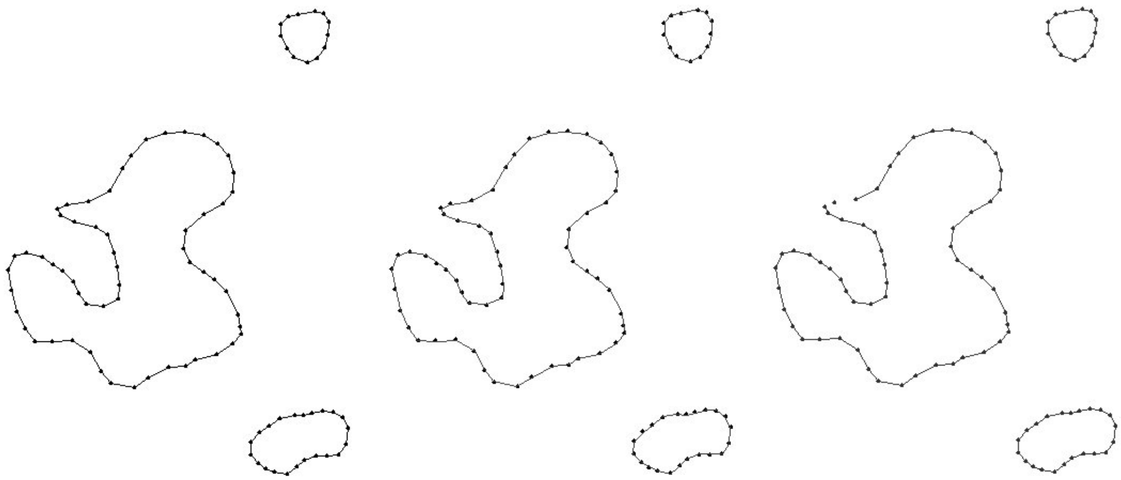
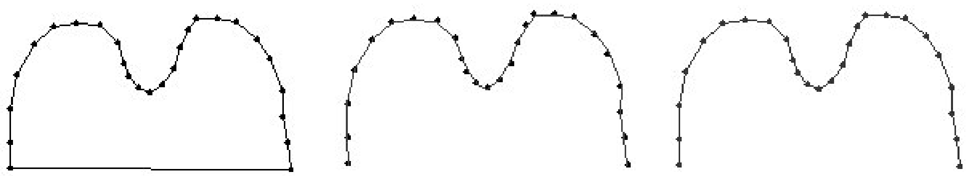

(c) Conservative Crust

Figure 12: Sample 7



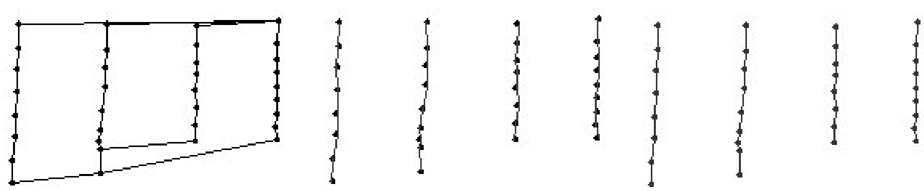
(a) NN-Crust (b) RNG (c) Conservative Crust

Figure 13: Sample 8



(a) NN-Crust (b) RNG (c) Conservative Crust

Figure 14: Sample 9



(a) NN-Crust (b) RNG (c) Conservative Crust

Figure 15: Sample 10