Reconstructing Curves with Sharp Corners

Tamal K. Dey and Rephael Wenger *

June 2, 2000

Abstract

In this paper we present a heuristic to reconstruct non-smooth curves with multiple components. Experiments with several input data reveals the effectiveness of the algorithm in contrast with the other competitive algorithms.

1 Introduction

Curve reconstruction is the problem of computing a piecewise linear approximation to a curve from a set of sample points. Applications include detecting boundaries in image processing, computing patterns in computer vision and intelligent systems, extracting information from aerial surveys in geographic information systems, and fitting a spline through a set of points in mathematical modeling. As a result of this vast application domain, the problem has drawn attention of researchers for a long time [5, 6, 9, 10, 14, 15]. Recently, renewed interest in the problem has focused on its relation to the more demanding problem of *surface reconstruction* in CAD applications [1, 3, 13]. Advances in laser technology have made it easier to obtain samples from the boundary of an object but these samples are useless without effective procedures to reconstruct the object surface from them. Curve reconstruction is the lower dimensional version of this problem and provides useful insights and experiences for designing these algorithms.

Obviously, unless samples from a curve are "dense enough," it is difficult, if not impossible, to reconstruct a close approximation to the original curve. Amenta, Bern and Eppstein [2] concretized the idea of "dense" sampling using the concept of feature size. The *medial axis* of a curve Γ is the set of points in the plane which have more than one closest point on Γ. The *feature size*, $f(p)$, of a point $p \in \Gamma$ is the distance between p and the closest point on the medial axis. This distance captures the features of the curve; $f(p)$ is small where Γ has small features and it is large where Γ is flat.

Amenta et. al. defined sampling density based on a parameter ϵ by requiring that each point $p \in \Gamma$ has a sample point within distance $\epsilon f(p)$. Several other algorithms have been developed with this assumption of sampling density [7, 8, 12]. This sampling density condition can be satisfied for *smooth* curves in practice. However, non-smooth curves with *corners,* i.e., points where the left and right tangents do not match, pose an intrinsic difficulty with the approach. The medial axis of such curves passes through the corners, and thus one faces the practical impossibility of sampling the curve infinitely near the corner to satisfy the sampling condition. In fact, the algorithms of [2, 7, 8, 12] do not work for curves with sharp corners as illustrated in Figure 2.

*Dept. of CIS, Ohio State University, Columbus, Ohio 43210. email: {tamaldey,wenger}@cis.ohio-state.edu

The first algorithm that successfully handled curves with corners is by Giesen [11]. He proved that the Traveling Salesman Tour reconstructs a curve with corners if the sampling density is higher than a certain threshold. This threshold depends upon the angles between left and right tangents at the corners. Building on this result Althaus and Mehlhorn showed how Giesen's algorithm can be made polynomial with linear programming under an improved sampling condition [4].

Although the Traveling Salesman Tour of a set of sample points reconstructs curves with corners, it has two drawbacks. First, it is not a good reconstruction of a curve with multiple components, since it connects these components by a single tour. Figure 1 shows an example illustrating this shortcoming of the Traveling Salesman Tour. Secondly, it is not clear if the reconstruction algorithms based on the Traveling Salesman Tour can be generalized to three dimensions. The natural generalization of the Traveling Salesman Tour is a minimal surface, i.e., a surface containing the sample points with minimal area. It is not known if such a minimal surface is a good approximation for a surface with sharp features and, if it is, whether it can be computed efficiently. These shortcomings of Traveling Salesman based algorithms motivated us to look for some other method that can handle nonsmooth curves.

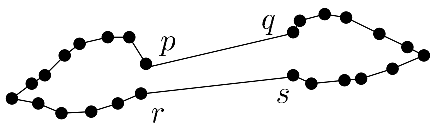

Figure 1: Traveling Salesman Tour connects two components through unwanted edges pq and rs.

Our algorithm is based on some new observations that we believe will be useful for further developments in the area. We do not have a theoretical proof of the guarantee of the algorithm, but experimental success leads to beliefs that such rigorous analysis may exist, possibly with some modifications. In Figure 2 we compare the output of our algorithm with those of the three other competitive algorithms: CRUST of [2], NN-CRUST of [7], CONSERVATIVE CRUST of [8]. It illustrates that our algorithm handles curves with sharp corners and multiple components quite effectively in comparison with other known algorithms. All other algorithms miss some of the necessary edges near sharp corners and some of them add extra edges. In particular, the points p and q near a sharp corner are joined by all algorithms except ours. Also, see Figures 11 and 12.

The paper is organized as follows. Section 2 provides some basic definitions. The algorithm is described in section 3 and experimental observations are detailed in section 4. We conclude in section 5.

2 Corners

As in [4, 11], we require that the sampled curve be planar and simple, i.e., not intersect itself, although it may have multiple components. Furthermore, *left tangents* and *right tangents* are defined everywhere and they are same at all points except at some isolated points called *corners*, where they make an angle less than π. The curve may be *closed*, or may have *endpoints*. Corners are isolated points and may not be in the sample set. We can only require that the neighborhood of a corner is densely sampled.

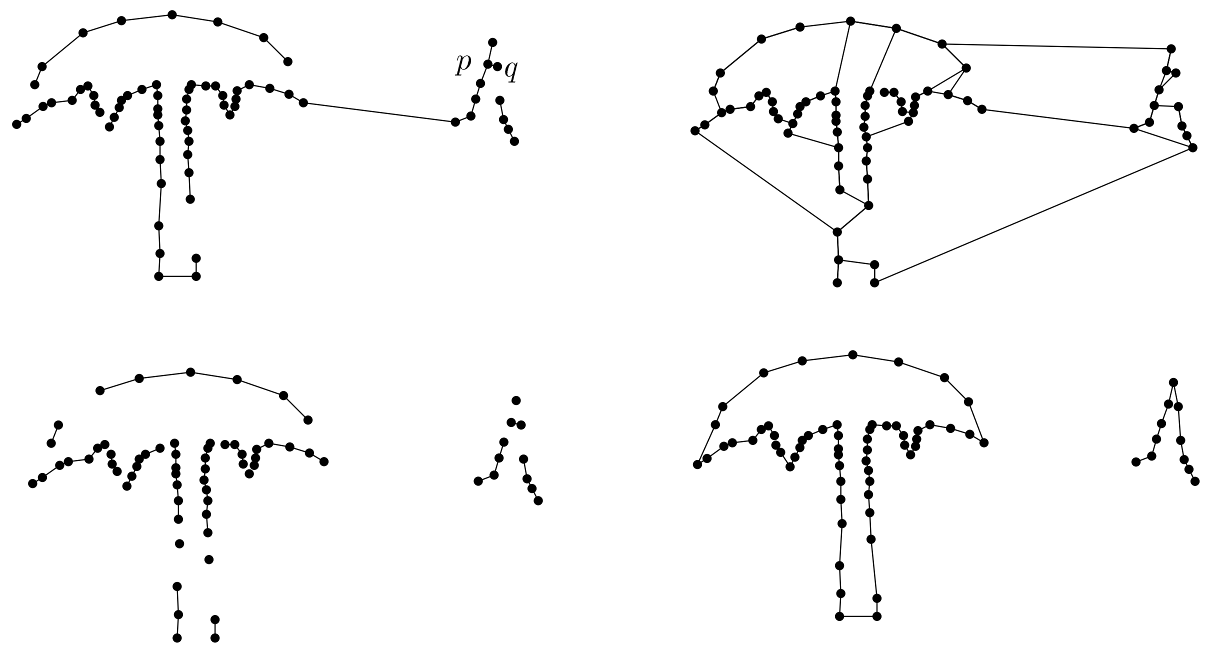

Figure 2: Output of CRUST (top left), output of NN-CRUST (top right), output of CONSERVATIVE CRUST (bottom left), output of our algorithm (bottom right). TSP tour, in this case, will join the two components of the curve.

Sample points partition a curve into arcs. Each sample point is adjacent to two such arcs. We call a sample point *regular* if its two adjacent arcs do not contain any corner or boundary point. A sample point is called *corner* if it is a corner point or at least one of its two adjacent arcs contains a corner point. A sample point is called *boundary* if it is an endpoint or if at least one of its two adjacent arcs contain a boundary point on the curve. We assume that the curve sampling is dense enough so that no sample point is adjacent to both a corner point and a boundary point. Given a sample of a curve, we say that an edge is *correct* if it connects two adjacent samples on the curve.

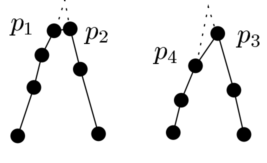

Figure 3: Corner samples.

Our algorithm relies heavily on the estimations of normals at the sample points. This is done by using "poles" in a Voronoi cell as introduced in [1]. It is comparatively easy to estimate the normals at the regular samples. Corner samples and boundary samples pose difficulty in normal estimation and thus are difficult to handle in reconstruction. However, the neighboring regular samples can be effectively used to detect correct edges that should be incident to the corner and boundary samples. What about edges that should connect two corner samples? There are two distinct cases as shown in Figure 3. The two corner samples p_1 and p_2 in the left picture behave differently than the two corner samples p_3 and p_4 in the right picture. The corner sample p_3 behaves like a true corner point on the curve. Our algorithm can estimate the normal at p_4 though it is a corner sample, whereas it is comparatively difficult to estimate the normals at both corner points p_1 and p_2. Consequently, the detection of p_1p_2 is harder than the detection of p_3p_4. Figure 10 in section 4 illustrates this aspect. Incorrect estimation of normals lead to incorrect edge computations. We will see that edges incorrectly joining a corner sample with other samples can be eliminated with a topological clean up in a postprocessing.

3 Algorithm

The algorithms of [2, 7, 8, 12] work on the assumption that the sample is sufficiently dense. All these algorithms require the following condition.

Condition (R): Any point on the sampled curve has a sample point within a distance of $\epsilon < 1$ times its distance from the medial axis.

As pointed out earlier, this sampling condition cannot be satisfied practically for non-smooth curves since it would require the curve to be sampled with infinite density near the corners. So, we require a different sampling condition near sharp corners.

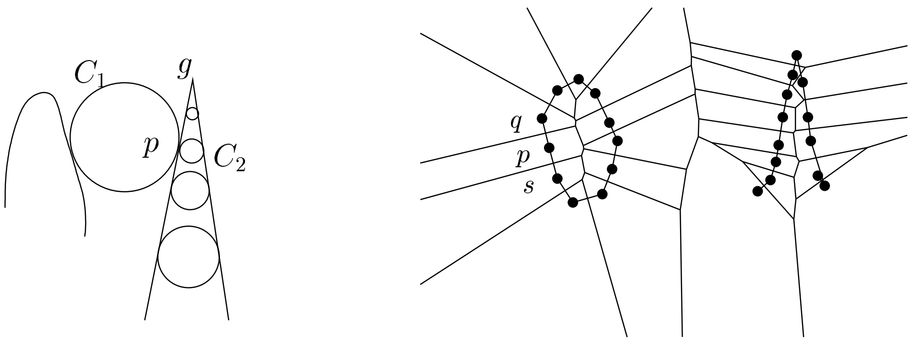

Figure 4: One of the two touching circles, C_1, C_2, is big near the corner g. Right picture shows elongated Voronoi cells on one side near the corner.

Let g denote a corner point on the curve Γ. At each smooth point $p \in \Gamma$ there are two maximal circles C_1 and C_2 that touch Γ tangentially at p and at least one other point, or they reach infinity. As p approaches g, the radius of one of these circles approaches zero. The sampling condition (R), in a sense, depends on the smaller of the two circles and thus requires infinite density near g. Instead, if we modify the sampling condition in a small neighborhood of corners so that it depends on the larger of the two circles, we do not face this problem. At the same time, the angle θ_g between the left and the reversal of the right tangent at g should also influence the sampling condition near g. The smaller the angle, the denser should be the sampling. These requirements are formalized in the condition (R$'$) below, which is similar to the one proposed in [4].

Consider growing a ball centering g as long as its boundary intersects the medial axis in a single point. Let r_g be the radius of this ball. Assume a *protective* ball centering g with radius cr_g for some suitable constant $c < 1$, say $c = 1/6$.

CONDITION (R$'$): Let p be any point on Γ. If p is in the protective ball of a corner g, it must have a sample point within $kr_g\theta_g$ distance; otherwise p must have a sample within $\epsilon f(p)$ distance, where $k < c < 1$ and $\epsilon < 1$ are suitable constants.

Intuitively, as the angle between the two branches of the curve at g grows smaller, it becomes more difficult to differentiate these branches and so the sampling should become denser. On the other hand, as the radius r_g of the protective ball grows larger, there is less interference from sample points which are on other parts of the curve, and so fewer samples are needed.

We do not use condition R$'$ explicitly since we do not provide a proof of correctness of our algorithm. However, the rationale behind the conditions that we use for selecting output edges can be explained by assuming (R$'$). Several experiments with different examples support these explanations.

Our algorithm is based on nearest neighbors that were used by Dey and Kumar in [7] for smooth curve reconstructions. Their algorithm computes two edges emanating from a sample, one connects it to the nearest neighbor, and the other connects it to the nearest neighbor in the opposite direction. So, if q is the nearest neighbor of p, then pq is computed and the other edge incident to p connects to s where s is the nearest neighbor with $\vec{ps}$ making an angle more than $90°$ with $\vec{pq}$; refer to the right picture in Figure 4. The motivation for this choice of edges comes from the fact that the two edges incident to p should make a large angle close to $180°$ and approximate the tangential direction at p to the sampled curve. It can be shown that the edges connecting nearest neighbors approximate the tangential direction at the sample points if the sampling satisfies condition (R). Unfortunately, this assertion does not hold for curves with corners. First of all, tangents are not defined at corners, and secondly condition (R) cannot be satisfied in practice near the corners. A crucial observation we make is that, it is still possible to estimate the tangents, or equivalently normal directions at the regular sample points under the modified sampling condition (R$'$).

Estimating normals

Amenta and Bern [1] observed that a Voronoi cell V_p for a sample p is elongated along the normal direction if the sampling condition (R) is satisfied. This important observation led to estimating the normals at samples for reconstructing surfaces. They introduced the concept of "poles" which are farthest Voronoi vertices from the samples in their respective Voronoi cells. If a Voronoi cell is bounded, then the line l_p through the sample and the pole estimates the normal up to orientation. Otherwise, the normal is estimated to be the average of the directions given by the two unbounded rays. For the case of curves, consider the two tangential circles C_1 and C_2 at a sample p that have centers on the medial axis. These circles are large compared to the edges pq and ps that are incident to p in the correct reconstruction. This follows from the sampling condition (R). The two circles C_1 and C_2 are empty of samples which implies that their centers lie in the Voronoi cell V_p. Thus V_p must lie within the narrow slab formed by the lines containing the dual Voronoi edges of pq and ps and contain the centers of C_1 and C_2. This forces V_p to be elongated on both sides of the curve at p along the normal direction. In the non-smooth case, at least one of C_1 and C_2 is large compared to the edges pq and ps under the modified sampling condition (R$'$). This implies that V_p is still elongated along the normal direction at p, but only on one side. See Figure 4. Thus, the normals can still be estimated using poles, or unbounded Voronoi edges. This is the first step of our algorithm.

3.1 Nearest Neighbors

Once we estimate the normals, we can follow the nearest neighbor strategy of [7]. For this, each sample point p can connect to the nearest neighbors on each side of the estimated normal line l_p at p. One needs to be careful though that the normals to these edges do not make large angles with l_p.

Angle condition

Figure 5 shows an example where the nearest neighbor captures a wrong edge pq with its normal making an angle close to 90° with the estimated normal at p (estimated normals are indicated with a small stick at the samples). To rectify this problem, we introduce the *angle condition*.

Angle Condition: An edge pq qualifies for the nearest neighbor test only if its dual Voronoi edge makes an acute angle less than a user defined parameter α with l_p.

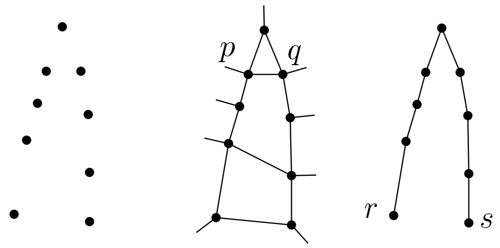

Figure 5: A sample (left), reconstruction without angle condition (middle) and reconstruction with angle condition (right). Small sticks at the samples indicate estimated normal direction.

Typically, we have observed that an angle between 35° and 40° is a good choice for α in most cases. For example, setting a maximum angle bound in this range disallows wrong edges in the output shown in Figure 5. There is another benefit provided by this angle condition. In some cases, it helps detecting the boundary points of an open curve. The samples r and s in Figure 5 are detected as boundary points due to this reason.

Ratio condition

The angle condition alone is not sufficient to discard wrong edges. Consider the example in Figure 6. The points p and r should not be joined with an edge. However, the dual Voronoi edge of pr makes a small angle with the estimated normal at p as does the correct edge pq. The nearest neighbor algorithm will pick up the edge pr over pq since r is closer to p than q. This anomaly results from the fact that the curve is undersampled with respect to the sampling condition (R) in the neighborhood of the corner.

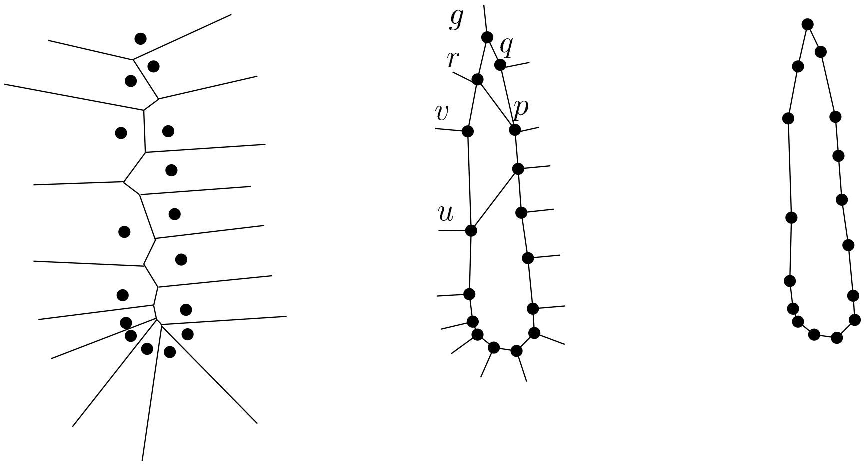

Figure 6: A sample with the Voronoi diagram(left), reconstruction with angle condition but not with ratio condition (middle), reconstruction with both angle and ratio condition (right).

To fix this problem we observe that the ratio of the lengths of the dual Voronoi edge and the edge pq is much larger than the same ratio for the edge pr. The intuitive explanation for this fact is that there is a much larger empty circle on the right side of the curve near p than the empty circle to the right of the curve near r. The sampling condition (R$'$) ensures that the edge length of pq is small compared to the empty circle to the right of Γ at p, but the same condition does not hold for r. Taking the cue from this observation we require that an edge satisfy the following *ratio condition*.

Ratio Condition: The ratio of the length of the dual Voronoi edge to the length of the edge is more than a preset threshold ρ.

We observe that the range $1.7 - 2.0$ works best for ρ in most cases.

3.2 Topological condition

The nearest neighbor algorithm chooses at most two edges per sample point. Nonetheless, some sample may acquire more than two edges due to other non-regular sample points. For example, the point p in Figure 7 has been connected with three edges, two of them being correct, and one is not. The edge pr is acquired incorrectly by the corner sample r. The estimated normal at r is incorrect. However, p being a regular sample has its two computed neighbors much closer than r. Thus, assuming that the input is sampled from a 1-manifold, i.e., curves without branchings, we can delete the longest edge pr incident to p. In general, we keep only the smallest two edges incident to a sample and delete others. Figure 7 shows an example before and after this pruning.

Below, we provide the pseudocode for our algorithm which we name Gathan, the sanskrit for "construction".

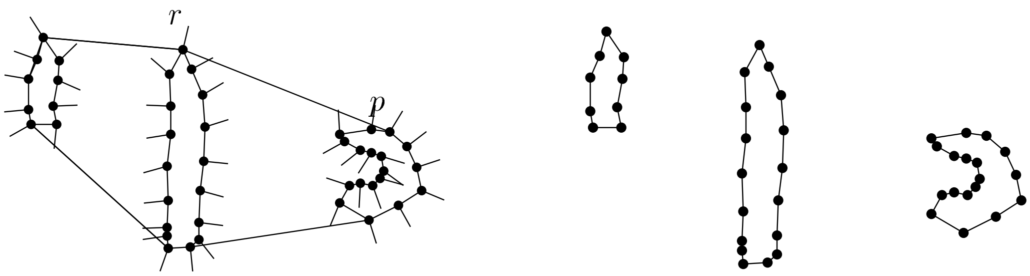

Figure 7: Reconstruction with angle and ratio conditions but not with the topology condition (left); reconstruction with angle, ratio and topology conditions (right).

GATHAN(P, α, ρ)
  Compute the Voronoi diagram V_P;
  for each $p \in P$ do
    Compute the pole and the normal line l_p.
    Let E be the set of Delaunay edges incident to p satisfying the following conditions:
    A. normal to each $e \in E$ makes an acute angle less than α with l_p,
    B. $h/\ell > \rho$ where ℓ and h are the lengths of e and its dual Voronoi edge.
    Keep only the smallest edges $pq \in E$ and $ps \in E$ on each side of l_p.
  endfor
  Delete any edge that is not among the smallest two edges incident to a sample point.
end

4 Observations

We experimented with different values of the two parameters, ρ and α. If we are strict on the angle condition, many correct edges may not qualify for output. On the other hand, if we increase the value of α, the algorithm allows edges whose normal makes larger angle with the estimated normal at the respective sample points. The nearest neighbor algorithm picks up wrong edges connecting two regular samples on the two legs from a corner point. We experimented with several examples, and found that the range $35^\circ - 40^\circ$ is the most suitable for most of the input. Figure 8 shows the effect of varying α over a range of $0^\circ - 90^\circ$.

We also experimented with the parameter ρ. If the value of ρ is small, the algorithm allows edges connecting regular samples on two different legs. These edges may not be eliminated solely by the angle condition. Figure 9 shows such an example. Wrong edges are eliminated with increasing value of ρ, but so are some of the correct edges. So, again we need to strike a balance. Experiments with several examples suggest that a value between 1.7 and 2.0 is appropriate for ρ. See Figure 9 for an illustration.

One good feature of the algorithm is that, in some cases, it connects samples correctly even at places where the sampling is relatively sparse. Figure 6 shows such an example. The edge uv is computed correctly though sampling is not very dense. The reason for this added feature is that the algorithm is based on a sampling which may be sparse if one side of the curve in the respective

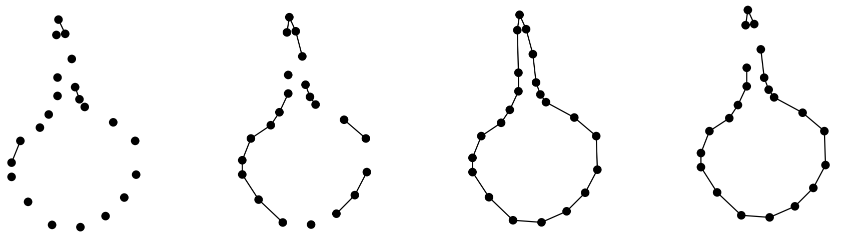

Figure 8: From left to right, reconstruction with $\alpha = 1°, 10°, 30°$ and $80°$.

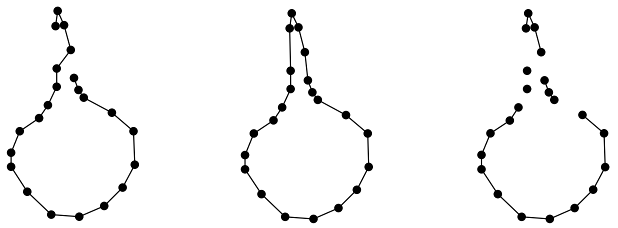

Figure 9: From left to right reconstruction with $\rho = 1, 2$ and 3.

region does not have medial axis nearby.

We also experimented with curves that have boundary points. Some cases may arise where the distinction between sharp and boundary samples is not obvious. For example, the point p in Figure 10(a) may be a boundary or a corner sample. In this ambiguity, the algorithm decides it to be a corner sample. On the other hand, the example in Figure 10(b) shows that the algorithm decides corner samples p, q to be boundary ones. However, obvious boundary points such as the ones shown in Figure 10(c) are well detected by the algorithm.

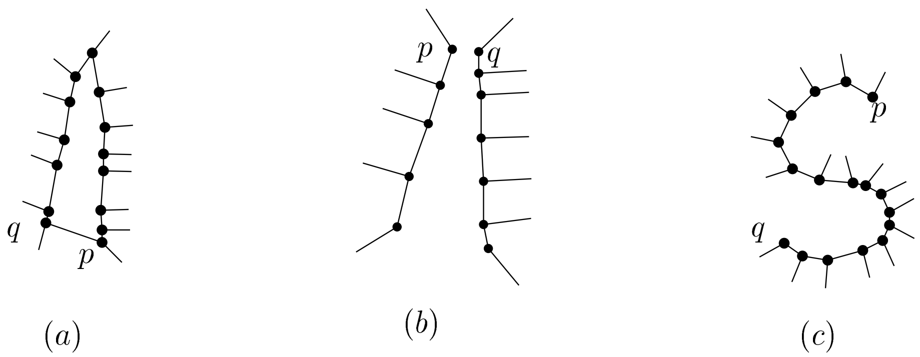

Figure 10: Ambiguity between boundary and corner samples.

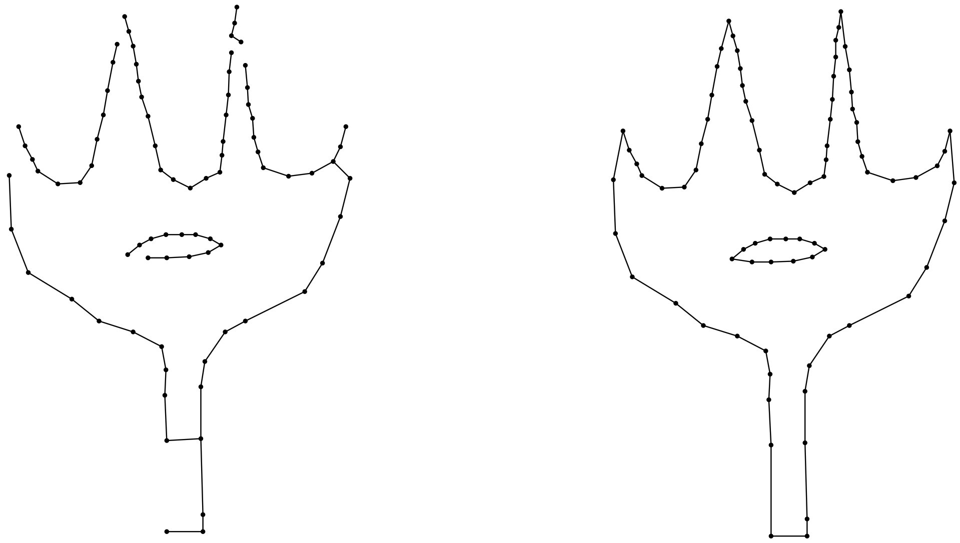

Figure 11: Output of Crust (left), output of Gathan (right).

5 Conclusions and future work

We presented new heuristics for curve reconstruction to handle smooth and nonsmooth curves with multiple components. The algorithm can also detect boundary points in many cases. We experimented with several inputs and our algorithm performed better than the existing algorithms near sharp corners. Two more examples are shown in Figures 11 and 12. In spite of its good performance we could not prove the theoretical guarantee for our algorithm. It remains open if it is possible to provide such a guarantee, or if some modifications are necessary to obtain such a guarantee. Currently, we are working towards that goal.

An obvious question is how to extend the algorithm to three dimensions to handle non-smooth surfaces. In practice, samples derived from non-smooth surfaces pose serious difficulty for reconstruction. All steps in our algorithm can be extended to three dimensions. However, it remains to be seen how well these heuristics perform in three dimensions. Currently, efforts are under way to make these extensions.

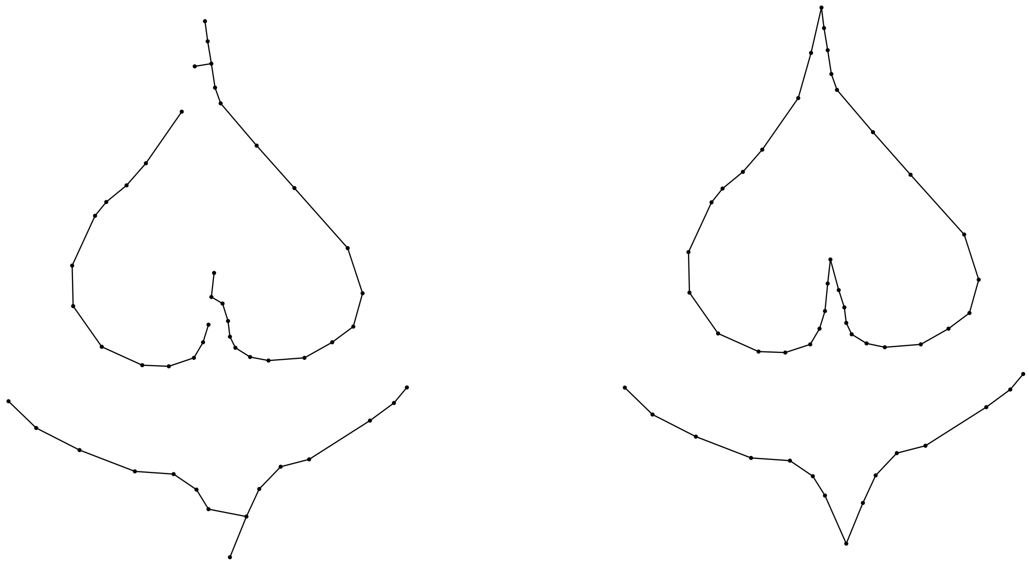

Figure 12: Output of Crust (left), output of Gathan (right).

References

[1] N. Amenta and M. Bern. Surface reconstruction by Voronoi filtering. *Discrete Comput. Geom.*, **22** (1999), 481–504.

[2] N. Amenta, M. Bern and D. Eppstein. The crust and the β-skeleton: combinatorial curve reconstruction. *Graphical Models and Image Processing*, **60** (1998), 125-135.

[3] N. Amenta, S. Choi, T. K. Dey and N. Leekha. A simple algorithm for homeomorphic surface reconstruction. To appear in *16th ACM Sympos. Comput. Geom.*, 2000.

[4] E. Althaus and K. Mehlhorn. Polynomial time TSP-based curve reconstruction. *Proc. 11th ACM-SIAM Sympos. Discrete Algorithms*, (2000), 686–695.

[5] D. Attali. r-regular shape reconstruction from unorganized points. *Proc. 13th ACM Sympos. Comput. Geom.*, (1997), 248–253.

[6] F. Bernardini and C. L. Bajaj. Sampling and reconstructing manifolds using α-shapes. *Proc. 9th Canadian Conf. Comput. Geom.*, (1997), 193–198.

[7] T. K. Dey and P. Kumar. A simple provable algorithm for curve reconstruction. *Proc. ACM-SIAM Sympos. Discr. Algorithms*, (1999), 893–894.

[8] T. K. Dey, K. Mehlhorn and E. A. Ramos. Curve reconstruction: connecting dots with good reason. *Comput. Geom. Theory Appl.*, **15**, (2000), 229–244.

[9] H. Edelsbrunner, D. G. Kirkpatrick and R. Seidel. On the shape of a set of points in the plane. *IEEE Trans. Information Theory*, **29**, (1983), 551-559.

[10] L. H. de Figueiredo and J. de Miranda Gomes. Computational morphology of curves. *Visual Computer*, **11**, (1995), 105–112.

[11] J. Giesen. Curve reconstruction, the TSP, and Menger's theorem on length. *Proc. 15th ACM Sympos. Comput. Geom.*, (1999), 207–216.

[12] C. Gold and J. Snoeyink. Crust and anti-crust: a one-step boundary and skeleton extraction algorithm. To appear in *Algorithmica*, 2000. Also *Proc. 15th. ACM Sympos. Comput. Geom.*, (1999), 189–196.

[13] H. Hoppe, T. DeRose, T. Duchamp, J. McDonald and W. Stuetzle. Surface reconstruction from unorganized points. *SIGGRAPH 92*, (1992), 71-78.

[14] M. Melkemi. $\mathcal{A}$-shapes and their derivatives. *13th ACM Sympos. Comput. Geom.*, (1997), 367–369.

[15] J. O'Rourke, H. Booth and R. Washington. Connect-the-dots: A new heuristic. *Comput. Vision Graph. Image Process.*, **39**, (1987), 258–266.