Reconstructing Curves with Sharp Corners

Tamal K. Dey and Rephael Wenger
Dept. of CIS, Ohio State University
Columbus, Ohio 43210
email: {tamaldey,wenger}@cis.ohio-state.edu

Abstract

In this paper we present a new algorithm for curve reconstruction that has multiple applications in image processing, geographic information systems, pattern recognition and mathematical modeling. The algorithm can deal with nonsmooth curves with multiple components that cannot be handled by existing algorithms. Experiments with several input data reveals the effectiveness of the algorithm in contrast with the other competitive algorithms for the problem. An attractive feature of the algorithm is that it is extendible to three dimensions for surface reconstructions.

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. In the past, the problem has been studied for its applications to pattern recognition[7, 8, 12]. Recently, renewed interest in the problem has focused on its relation to the more demanding problem of *surface reconstruction* in CAD applications [1, 4, 11]. 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 from p to the closest point on the medial axis. This distance captures the features of the curve; $f(p)$ is small where Γ is "narrow" 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 [5, 6, 10]. This sampling density condition can be satisfied for *smooth* curves in practice. However, nonsmooth curves with *corners,* i.e., points whose 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 is required to sample the curve infinitely near the corner to satisfy the sampling condition. In fact, the algorithms of [2, 5, 6, 10] do not work for curves with sharp corners. See Figure 1.

The first algorithm that successfully handled curves with corners is by Giesen [9]. 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 right and left tangents at the corners. Althaus and Mehlhorn gave a polynomial time version of Giesen's algorithm based on linear programming[3].

Although the Traveling Salesman Tour of a set of sample points reconstructs curves with corners, it has

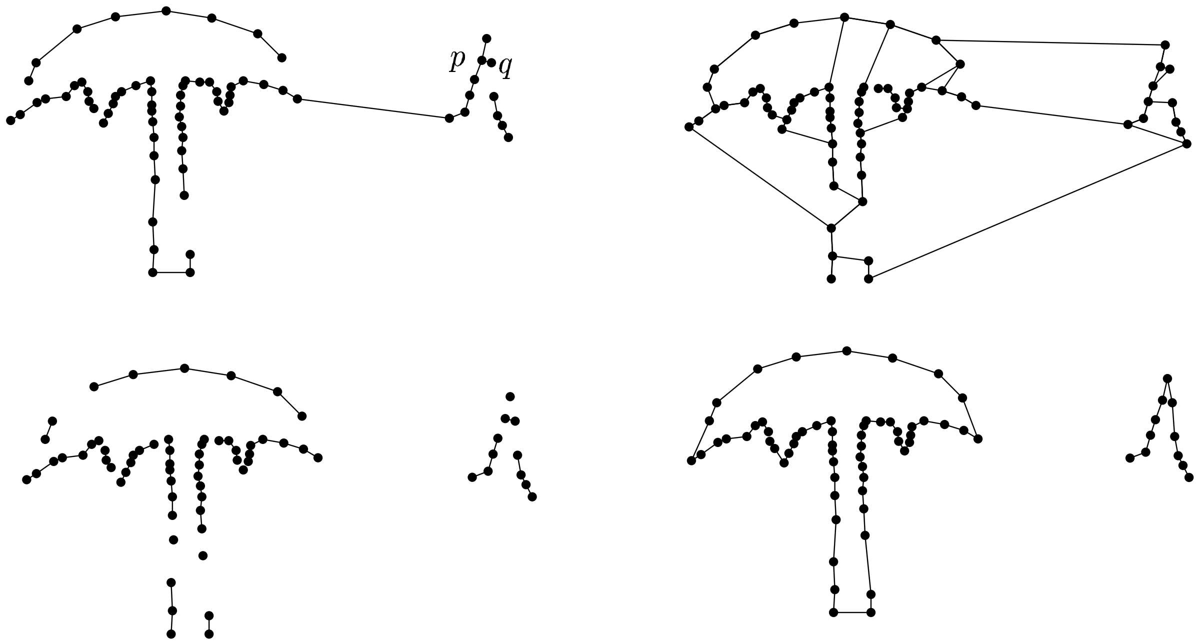

Figure 1: Output of Amenta-Bern-Eppstein algorithm (top left), output of Dey-Kumar algorithm (top right), output of Dey-Mehlhorn-Ramos algorithm (bottom left), output of our algorithm (bottom right). TSP tour, in this case, will join the two components of the curve.

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 2 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 generalize 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 multiple components of nonsmooth curves and could be easily extended to three dimensions.

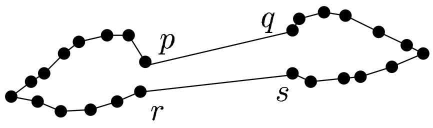

Figure 2: Traveling Salesman Tour connects two components through unwanted edges pq and rs.

In practice, samples are derived from curves that have multiple components, sharp corners and boundaries. A real example from GIS application in [10] shows all these possibilities in a single sample set. An ideal reconstruction algorithm should handle all of these possibilities. In this paper we present such an algorithm and show evidence of its effectiveness on several examples. In Figure 1 we show the output of the crust algorithm of [2], nearest neighbor algorithm of [5], conservative crust algorithm of [6] and our algorithm on a two component curve. It illustrates that our algorithm handles curves with sharp corners, boundary points and multiple components quite effectively in comparison with other algorithms. All other algorithms miss some of the necessary edges near sharp corners and some of them put extra edges. In particular, the points p and q near a sharp corner is joined by all algorithms except ours. Also, see Appendix A. Our algorithm is based on several new observations that we believe will be useful for further developments in the area. We do not provide a theoretical proof of the guarantee of the algorithm, but experimental success leads to beliefs that there might exist such rigorous analysis.

2 Corners

As in [9, 3], we require that the sampled curve be planar and do not have crossings (multiple points). Further, *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 can be *closed*, or may have *endpoints*. Corners are difficult to sample. They are isolated, and stipulating that a particular point be sampled is impractical. We can only require that the neighborhood of corners are densely sampled.

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 corners or boundary point. A sample point is *corner* sample 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.

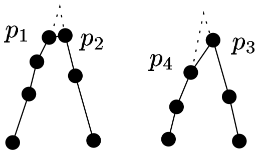

Figure 3: Corner samples.

We will see that it is comparatively easy to estimate normals to the curve at 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 with the corner and boundary samples. But, 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 very 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. We will see that edges incorrectly joining a corner sample with other samples can be eliminated with a topological criterion in a postprocessing.

3 Algorithm

The algorithms of [2, 5, 6, 10] works on the assumption that the sample is sufficiently dense. Precisely, all these algorithms require the following condition.

Condition (S): Any point on the sampled curve has a sample point within a distance that is $\epsilon < 1$ times smaller than its distance from the medial axis.

As pointed out earlier, this sampling condition cannot be satisfied for nonsmooth curves since it would require the curve to be sampled with infinite density near the corners. So, we require a modified sampling condition.

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 touches Γ tangentially at p and at least one other point, or they reach infinity. Near g, one of these circles gets smaller and smaller till it vanishes at g. This phenomenon does not happen if there is no corner. The sampling condition (S), in a sense, depends on both of these circles and thus require infinite density near g. Instead, if we modify the sampling condition in the neighborhood of corners as (S′) below we do not face this problem.

Condition (S′): Any point p on Γ must have a sample point within $\epsilon < 1$ times the radius of the larger circle between C_1 and C_2.

The main observation here is that the bigger of the two circles does not depend on the portion of the medial axis going through g. The rationale behind the conditions that we use for selecting output edges can be explained by assuming (S′). Several experiments with different examples support these explanations.

Our algorithm is based on nearest neighbors that were used by Dey and Kumar in [5] 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.

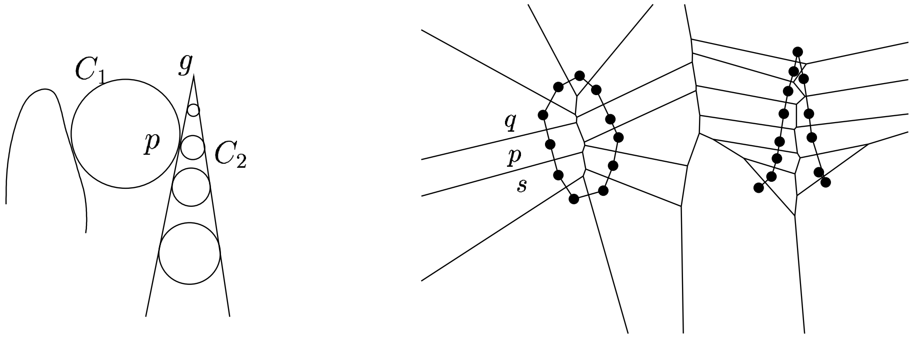

Figure 4: One of the two touching circles, C_1, C_2, is big near the corner g. Right picture shows elongated Voronoi cells on one side near the corner.

So, if q is the nearest neighbor of p, then pq is computed and the other edge incident with p connects to s where s is the nearest neighbor with $\vec{ps}$ making an angle more than 90° with $\vec{pq}$. See the right picture in Figure 4. The motivation for this choice of edges comes from the fact that the two edges incident with 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 (S). Unfortunately, this assertion does not hold for curves with corners. First of all, tangents are not defined at corners, and secondly condition (S) cannot be satisfied in practice near the corners. The most crucial observation we make is that, it is still possible to estimate the tangents, or equivalently normal directions at the sample points under the modified sampling condition (S$'$).

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 (S) 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 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. To see the reason on 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 with p in the correct reconstruction. This follows from the sampling condition (S). 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 nonsmooth case, at least one of C_1 and C_2 is large compared to the edges pq and ps under the modified sampling condition (S$'$). 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

We can follow the nearest neighbor strategy of [5] after estimating the normals at the sample points. For this, each sample point p can connect to the nearest neighbors on each side of the estimated normal line v_p through p. But, one needs to be careful so that these edges do not make large angles with v_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 (the normal is 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 angle less than a user defined parameter α with the estimated normal at p.

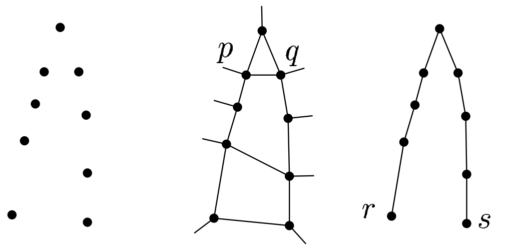

Figure 5: A sample (left), reconstruction without angle condition (middle) and reconstruction with angle condition (right). Small sticks at the samples indicate estimated normal direction.

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 good effect of 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. The trouble is again caused by the nonsmoothness at corner points. Consider the example in Figure 6. The points p and r should not be joined by an edge. But, the dual Voronoi edge of pr makes 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 we are using the sampling condition (S$'$) in the neighborhood of the corner. If instead we used the sampling condition (S) , then we would not have this problem, but the samples would have to become infinitely dense in the neighborhood of the corner.

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 (S$'$) 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.*

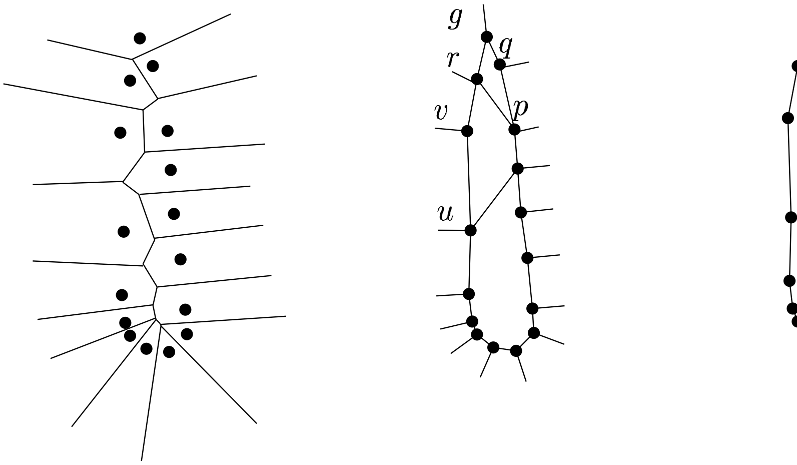

Figure 6: A sample with the Voronoi diagram(left), reconstruction with angle condition but not with ratio condition (middle), reconstruction with both angle and ratio condition (right).

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 some other 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 point r. The corner sample r behaves as a corner point and the estimated normal at r is incorrect. But, 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

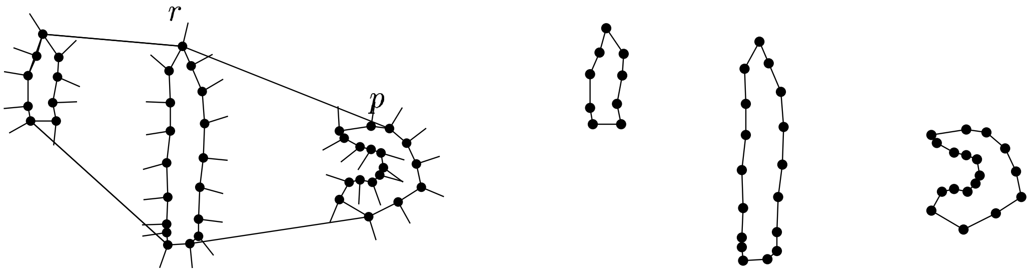

Figure 7: Reconstruction with angle and ratio conditions but not with the topology condition (left); reconstruction with angle, ratio and topology conditions (right).

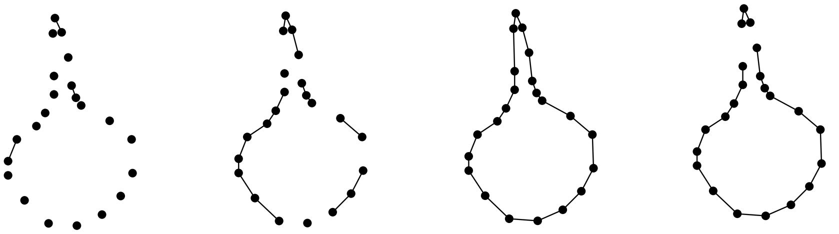

Figure 8: From left to right, reconstruction with $\alpha = 1°, 10°, 30°$ and $80°$.

delete the longest edge pr incident with p. In general, we keep only the smallest two edges incident with a sample and delete others. Figure 7 shows an example before and after this pruning. We name our algorithm GATHAN, the sanskrit for "construction". (See Figure 9.)

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 dual Voronoi edges make 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 $35° - 40°$ is the most suitable for most of the input. Figure 8 shows the effect of varying α over a range $0° - 90°$.

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 10 shows such an example. Increasing the value of ρ eliminates these faulty edges, but that may affect some of the correct edges connecting regular points. 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 10 for an illustration.

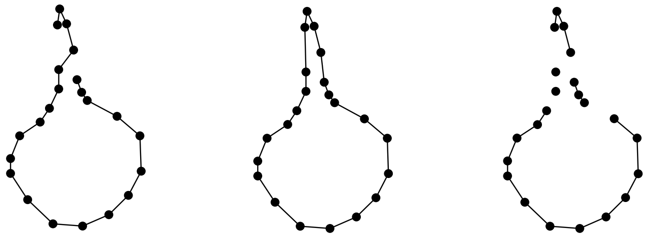

Figure 10: From left to right reconstruction with $\rho = 1, 2$ and $\rho = 3$.

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

```
GATHAN($P, \alpha, \rho$)
        Compute the Voronoi diagram $V_P$
        for each $p \in P$ do
            compute the pole and the normal line $v_p$
            Let $E$ be the set of Voronoi edges satisfying the following conditions:
            A. Each $e \in E$ makes an acute angle less than $\alpha$ with $v_p$.
            B. If $px$ is the dual Delaunay edge and $\ell$ is the lenght of $e$, the ratio $\ell/|px|$ is greater than $\rho$.
            Let $T$ denote the dual of $E$. Keep only the smallest edges $pq \in T$ and $ps \in T$ on each side of $v_p$.
        endfor
        Delete any edge that is not among the smallest two edges incident with a sample point.
end
```

Figure 9: GATHAN

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 region does not have medial axis nearby.

The algorithm is capable of detecting boundaries. Figure 11 shows an example where the boundary has been detected correctly. However, some cases may arise where the requirements between the sharp corners and the boundaries may conflict. Consider the example in the right picture of Figure 11. The edge between the points p and q may be added to detect a sharp corner at p, or it might be dropped to detect p as a boundary point. The algorithm decides in favor of the first in most cases.

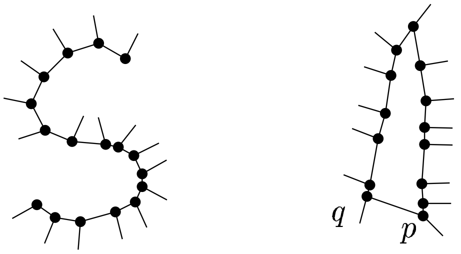

Figure 11: Boundary samples are detected in the left picture; in the right picture there is an ambiguity as to whether p is a boundary sample or a corner sample.

5 Conclusions and future work

We presented a new algorithm for curve reconstruction that can handle smooth and nonsmooth curves with multiple components. The algorithm can detect boundary points as well in many cases. No such algorithm is known till date. We experimented with several inputs and our algorithm performed better than the existing algorithms near sharp corners in all cases. Two more examples are shown in Appendix A. The algorithm is based on several new observations some of which are fundamental and may throw light on the difficult problem of reconstructing nonsmooth surfaces from samples.

All steps in our algorithm can be extended to three dimensions. The "poles" can be computed for each Voronoi cell and the angle condition can be checked with the dual Voronoi edges of the triangles and the estimated normals at the sample points. The ratio condition translates to the ratio of the length of the dual Voronoi edge to the radius of the circumcircle of the triangles. The topology criterion would require that each edge has no more than two triangles and each vertex is incident to a set of triangles that realizes a disk or a half-disk. Currently, efforts are being made to develop the software for nonsmooth surface reconstruction in three dimensions.

References

[1] N. Amenta and M. Bern. Surface reconstruction by Voronoi filtering. *Proc. 14th ACM Sympos. Comput. Geom.*, (1998), 39–48.

[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] E. Althaus and K. Mehlhorn. TSP-based curve reconstruction in polynomial time. To appear

in *Proc. ACM-SIAM Sympos. Discrete Algorithms*, 2000.

[4] C. Bajaj, F. Bernardini and G. Xu. Automatic reconstruction of surfaces and scalar fields from 3D scans. *SIGGRAPH 95*, (1995), 109–118.

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

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

[7] 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.

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

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

[10] C. Gold. Crust and anti-crust: a one-step boundary and skeleton extraction algorithm. *Proc. 15th. ACM Sympos. Comput. Geom.*, (1999), 189–196.

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

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

Appendix A.

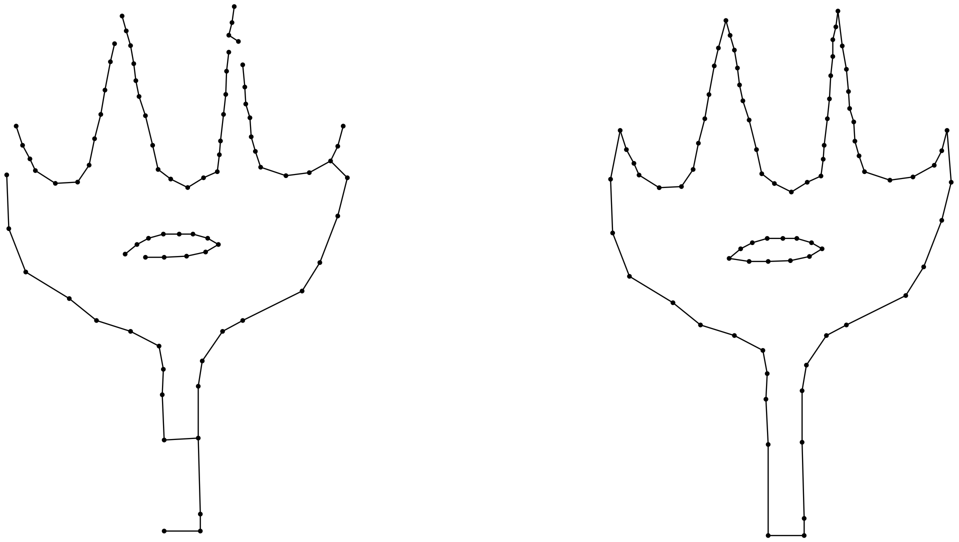

Figure 12: Crust algorithm(left), ours (right).

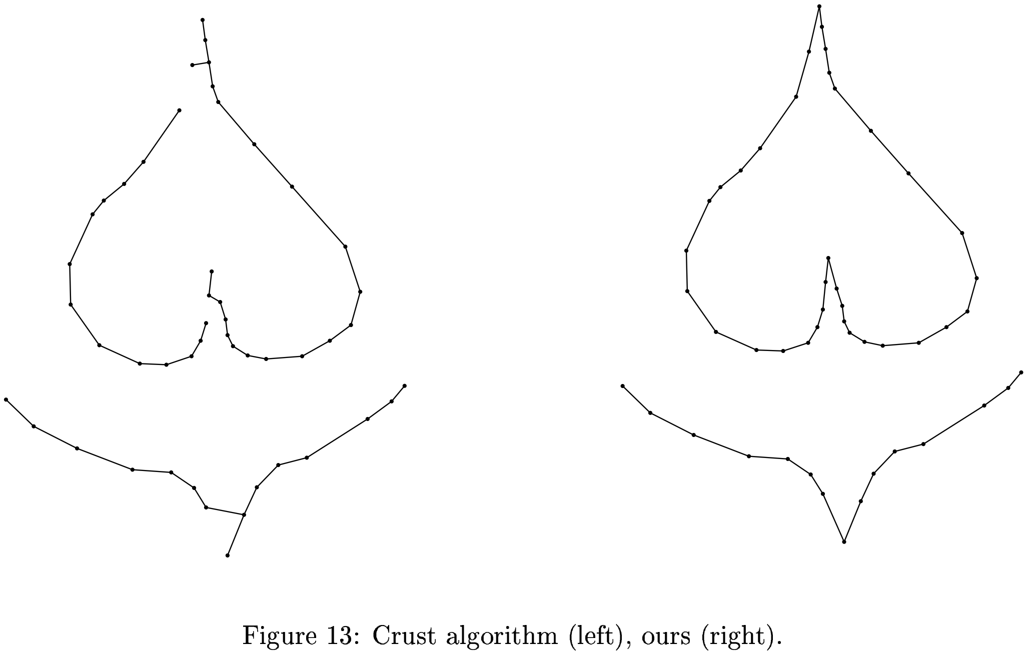

Figure 13: Crust algorithm (left), ours (right).