Reconstructing a Collection of Curves with Corners and Endpoints

Stefan Funke [*] *Edgar A. Ramos* [†]
Max-Planck-Institut für Informatik
Stuhlsatzenhausweg 85, 66123 Saarbrücken, Germany
{funke,ramos}@mpi-sb.mpg.de

Abstract

We present an algorithm which provably reconstructs a collection of curves with corners and endpoints from a sample set that satisfies a certain sampling condition. The algorithm outputs a polygonal reconstruction that contains the edges in the correct reconstruction of the curves and such that any additional edge between sample points is justified. Furthermore, we show that for any such collection of curves, there exists a sample set such that a slightly modified version of our algorithm outputs *exactly* the correct reconstruction. The algorithm also performs quite well in practice.

1 Introduction

We consider a collection of disjoint curves, each with a well-defined tangent at all but a finite number of points at which there is either an *endpoint* or a *corner* (at which the tangents from both directions exist but are different). See Figure 1.

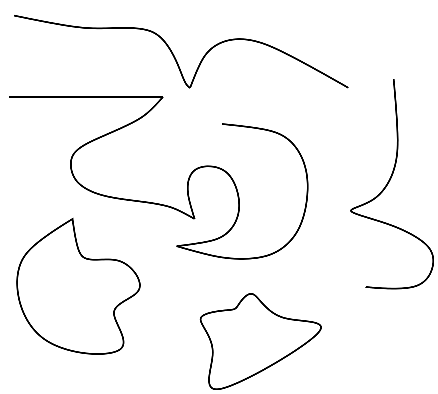

Figure 1: A collection of curves with corners and endpoints.

Given a set S of sample points (samples) from a collection Γ of open and closed curves, curve reconstruction is the problem of computing the graph $G(S, \Gamma)$, called the *correct reconstruction*, whose vertex set is S and that has an edge between two samples if and only if these samples are adjacent on a curve in Γ.

Obviously, it is not possible to correctly reconstruct a given curve from an arbitrary sample set from it. Therefore, some restrictions are needed on the sample which specify how dense a sampling has to be to guarantee a correct output of the algorithm. The first algorithms for curve reconstruction [3, 9, 10, 13] imposed a *uniform* sampling condition as they basically demanded that the distance between any two adjacent samples must be less than some constant. This is not satisfactory as it may require a dense sampling in areas where a sparse sampling is sufficient.

Amenta, Bern, and Epstein [2] introduced the concept of the *local feature size*. The local feature size $\mathrm{lfs}(p)$ of a point p on the curve is defined as the distance to the *medial axis* of this curve. The medial axis of a curve is defined as the set of points in the plane which have more than one closest point on the curve. Roughly, a neighborhood of a point of size equal to its local feature size is intersected by the curves in a single piece that winds up only a small angle.

Using this concept of local feature size, Amenta *et al.* define a non-uniform sampling condition that allows for sampling of variable density. Concretely, their sampling condition states that for any point $p \in \Gamma$, there must be a sample within distance $\epsilon \cdot \mathrm{lfs}(p)$. Then they give an algorithm that, given a sample set from a collection of smooth closed curves which satisfies this sampling condition for a certain $\epsilon < 1$, computes the correct reconstruction. This algorithm works by computing the Delaunay triangulation of the point set and then filtering it to obtain the reconstruction. A survey of algorithms based on *Delaunay filtering* can be found in [8]. Subsequently, several variations that still only handle smooth closed curves were presented [4, 12]. Later, Dey, Mehlhorn, Ramos [5] extended this work to handle a collection of open and closed smooth curves. Their algorithm is also based on Delaunay filtering.

The sampling condition with respect to the local feature size can be fulfilled for smooth curves. Problems arise, though, if the curves have corners, *i.e.* points for which left and right tangent disagree. In this case, the medial axis actually reaches the corners and, hence, requires an infinite

[*] The author was supported by a scholarship of the Deutsche Forschungsgesellschaft

[†] This research was partially supported by ESPRIT LTR project 28155 (GALIA)

dense sampling in the corner areas. The algorithms for *smooth* curves presented so far not only fail theoretically but also in practice (see for example the Figures at the end of this paper).

Giesen [11] uses a different sampling condition for corner areas and shows that for a sufficiently dense sampling, the TSP (traveling salesman problem) tour is the correct reconstruction for a single closed curve (possibly with corners). Althaus and Mehlhorn [1] have extended this result by showing that in this case the TSP tour can be computed in polynomial time. The problem with this approach is that so far it can only handle *single*, *closed* curves. Recently, Dey and Wenger [6] gave an algorithm that allegedly handles well corners and endpoints in practice. The algorithm has no guarantee and, in fact, it is not difficult to find counterexamples where it fails.

In this paper, we present an algorithm that provably reconstructs a collection of curves with endpoints and corners. As in [5], when the sampling condition only establishes a lower bound on the sampling density, it is not possible always to output the correct reconstruction. Instead, we verify that the output of the algorithm contains the correct reconstruction and that any extra edges in the output are justified. Furthermore, we present an upper bound on the sampling so that the algorithm outputs exactly the correct reconstruction. Our sampling conditions are stated with respect to the correct reconstruction of the sample set rather than with respect to the original curve (as in the case of the condition based on the medial axis), this allows to produce a correct reconstruction when it is possible from the samples even if it is not a faithful approximation of the curve. Because sampling conditions based on the medial axis have been widely used, we also make the following connection: If a sample is valid with respect to the medial axis, then it is also valid with respect to our sampling condition. Our algorithm is also based on Delaunay filtering. However, unlike all the previous algorithms, our filtering is not a simple local rule: The algorithm first detects 'smooth' edges reliably and then, starting from the endpoints of the resulting smooth chains, it 'walks' into the corners.

Concurrently with our work, Dey and Wenger have developed an algorithm based on their work in [6] which provably reconstructs a collection of closed curves with corners (no open curves) [7]. They also use the idea of first detecting 'smooth' areas and then exploring corner areas. While we have described our algorithm mostly in terms of empty β-balls, they use what they call the 'ratio condition' which denotes the ratio between the lengths of Delaunay edges and their dual Voronoi edges. In fact these concepts are quite the same. The conditions under which they guarantee correct exploration of corners differ from ours.

2 Sampling Condition

The problem of curve reconstruction as stated in the introduction only postulates that we connect all samples which are adjacent on the original curve. The sampling condition with respect to the medial axis does more, though. It also makes sure that the sampling (and the reconstruction) does not miss a single feature of the original curve. So it even guarantees that we obtain a good *approximation* to the curve. The question is whether we really want that. See for example Figure 2.

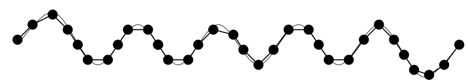

Figure 2: Wiggling Curve with MA sampling condition

If we apply the medial axis sampling condition, we are forced to sample this curve very densely as the 'wiggling' implies a very small local feature size for any point of the curve, and the correct reconstruction of the sample set is a pretty good approximation of the original curve not missing a single 'wiggle'. But if the only thing that we want is connecting the right samples (as stated in the original problem of curve reconstruction), a far less dense sampling should also do, as can be seen in Figure 3.

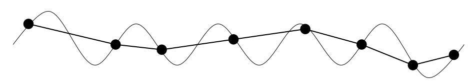

Figure 3: Wiggling Curve with less dense sampling

Of course, now the correct reconstruction does not approximate the curve as with the higher sampling density, but nevertheless, a correct reconstruction should be possible as well.

So what we propose is the following: The decision of whether a given sample set S from a collection of curves Γ is valid should not be stated with respect to the original curve, but rather with respect to the correct reconstruction $G(S, \Gamma)$ of this sample set. Such a sampling condition would allow to 'skip' details of a curve if this does not affect the possibility to correctly reconstruct the curve (in the sense of our original definition of the curve reconstruction problem).

As mentioned in the introduction, one problem with a sampling condition with respect to the medial axis is the fact that for curves with corners, the medial axis passes through the corners, hence requiring a infinitely dense sampling near corners. This can be fixed by relaxing the sampling condition within controlled areas around the corners. We will use this idea for our sampling condition in the next section as well.

We believe that a sampling condition expressed with respect to the correct reconstruction $G(S, \Gamma)$ is more sensible than a sampling condition expressed directly with respect to the curve (and the medial axis). However, since the medial axis sampling condition has become quite a standard in recent work on curve reconstruction, we will also show that our sampling condition is implied by the medial axis condition, *i.e.* all sample sets of a curve that are valid with respect to some medial axis sampling condition are also valid with respect to our sampling condition, i.e. our sampling condition is strictly weaker.

2.1 Our Sampling Condition Our sampling condition describes how the correct reconstruction $G(S,\Gamma)$ of a sample set S with respect to a collection Γ of open and closed curves (possibly with corners) must look like to guarantee certain properties of the output of our algorithm. Let us first consider a collection of open and closed curves without corners:

Sampling Condition for Smooth Areas: A sample set S for a collection of open and closed smooth curves Γ is valid, if for every edge $e = (p,q)$ of the correct reconstruction $G(S,\Gamma)$ the following holds (also see Figure 4):

- the two closed balls B_1, B_2 of radius $r_1 = \beta \cdot |e|/2$ touching p and q are empty of other samples
- the samples within the closed diametral ball around e of radius $r_3 = f_{dia} \cdot |e|$ can be connected in one chain that makes a total turn of less than θ_{ball}

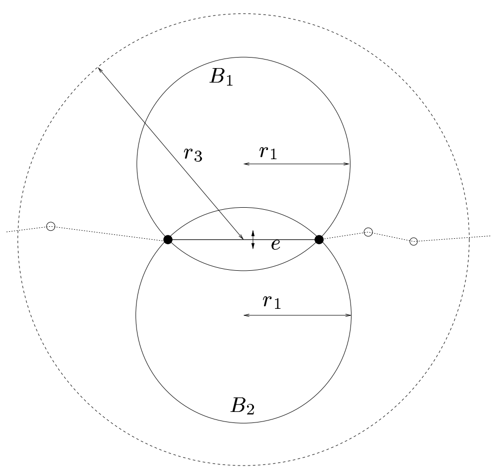

Figure 4: Sampling Condition for smooth areas

As with the medial axis sampling condition, the problems arise near corners. To solve them, we first have to identify areas near a corner and then define a weaker sampling condition for edges of the correct reconstruction which are completely contained in such a corner region.

Identification of corner areas: For each corner grow a ball around the corner point as long as:

- the ball intersects the curves in Γ in two smooth curve segments, which we refer to as the *legs* of the corner, each with an endpoint in the corner and the other on the boundary of the ball
- on each leg the tangent varies by at most θ_{slope}
- for any points s_1, s_2 on different legs, the line segment $s_1 s_2$ intersects the *interior* of the corner, that is, the area of the ball on the same side of the curve as the smaller angle at the corner [1]

The maximal ball obtained by this is then shrunken by a factor of f_{shrink}. The area within the shrunken ball defines a corner area.

Now we have to state a weaker sampling condition for these corner areas. First we drop the second part of the sampling condition for smooth areas of the curve, and allow samples on the other corner leg in the β-ball touching an edge from 'inside'. Additionally, we add a condition that makes sure that we can locally decide to which leg a sample belongs to. Otherwise it is difficult to decide locally whether there is a corner or not as it can be seen in Figure 5 (unless, we make further assumptions, *e.g.* that there is only a single closed curve).

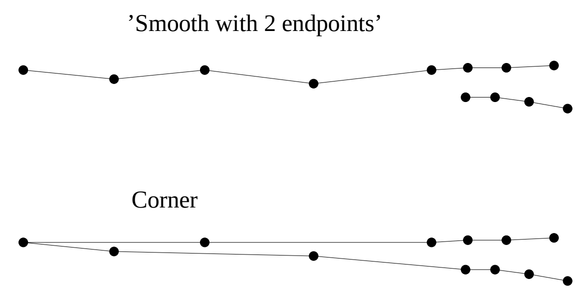

Figure 5: How to connect these samples ?

So our sampling condition for edges of the correct reconstruction *completely* inside a corner region is stated as follows:

Sampling Condition for Corner Areas: Let $e = (p,q)$ be an edge of the correct reconstruction completely inside a corner region, then we postulate (also see Figure 6):

- the closed ball B_1 of radius $\beta \cdot |e|/2$ touching p and q from the 'outside' of the corner is empty of other samples
- the closed ball B_2 of radius $\beta \cdot |e|/2$ touching p and q from the 'inside' of the corner may only contain other samples of the opposite leg ending in that corner

[1] This is equivalent to the corner point being on the boundary of the convex hull of the legs inside the ball.

(with the exception of the edge connecting the last two samples of each leg, whose inner β-ball may contain samples of both legs) which are inside the unshrunken corner ball

- the closed ball b_1 of radius $\beta_{low} \cdot |e|/2$ touching p and q from the 'outside' of the corner is empty of other samples

- the turn between e and its adjacent edges in the correct reconstruction must be less than θ_{turn} (again with the exception of the edge connecting the last two samples of each leg, but including the 'virtual' edges connecting those last two samples to the actual corner point)

- let l_{e_1}, l_{e_2} be the supporting lines of two edges e_1 and e_2 which are on different legs; then we have that the intersection point $I = l_{e_1} \cap l_{e_2}$ does not lie in opposite direction to the corner with respect to *both* e_1 and e_2

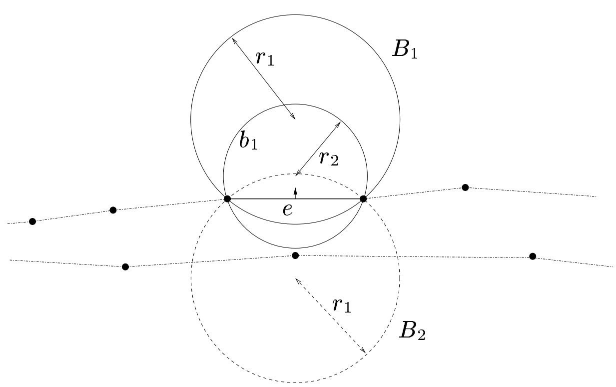

Figure 6: Sampling Condition for corner areas

We can now summarize our sampling condition.

General Sampling Condition: A sampling S for a collection of open and closed curves Γ (possibly with corners) is valid if for any edge $e \in G(S, \Gamma)$

- Sampling Condition for Smooth Areas holds if e is (at least partly) outside a corner region
- Sampling Condition for Corner Areas holds if e is completely inside a corner region
- Any smooth component of the correct reconstruction consists of at least 3 samples

Before we proceed to present the algorithm in the next section, we list some notation and conventions that will be used throughout the paper.

Notation/Conventions

corner sample: We call the last samples of each leg *corner samples*. If there is a sample close to the actual corner, we may say that there is only one corner sample as this sample fits into both legs.

corner spanning edge: We call the edge connecting the last samples of each leg *corner spanning edge*.

smooth/corner area: We call the area in the shrunken corner ball *corner area*, the rest is called *smooth area*.

smooth/corner edge: We call an edge which lies completely inside the shrunken corner ball a *corner edge*, the other edges are *smooth edges*

red/blue edge: We say an edge e is *red*, if it has two empty β-balls, we say e is *blue*, if it is not red, but has empty β- and β_{low}-balls on one side. [2]

In the figures, we draw a small normal arrow at the midpoint of an edge to indicate that it has β- and β_{low}-balls on that side. So every red edge has two such arrows, whereas every blue edge has only one arrow, as can be seen in Figures 4 and 6.

θ_{beta}, θ_{low}: To simplify notation we write θ_{beta} for $\arcsin \frac{1}{\beta}$ and θ_{low} for $\arcsin \frac{1}{\beta_{low}}$

2.2 Medial axis sampling condition implies our sampling condition First we identify the corner areas in the same way as for our sampling condition but with $\theta_{slope} = \theta_{turn}$. We then define a new local feature size $\text{lfs}'(p)$ for any point p on the curve as follows:

- if p is not in any shrunken corner ball, $\text{lfs}'(p)$ is equal to the distance of p to the medial axis of Γ

- if p is contained in a shrunken corner ball, $\text{lfs}'(p)$ is equal to the distance from p to the medial axis of the collection of curves obtained by removing the leg not containing p within the unshrunken corner ball [3].

The sampling condition is then stated as follows:

Sampling Condition w.r.t. the medial axis :

- for every point p on the curves there must be a sample within distance $\epsilon \cdot \text{lfs}'(p)$

[2]Clearly all smooth edges must be red, but some of the corner edges and even some edges that do not belong to the correct reconstruction might also be red.

[3]We define the local feature size of the corner point as the minimum distance to either medial axis

- for any edge $e = (p, q)$ of the correct reconstruction and any other sample r within the shrunken corner ball, the angle determined by e at r is less than $\theta_{angle} = \pi - \theta_{low}$
- any component of the collection of curves must contain at least 3 samples

COROLLARY 2.1. *Any sample set S valid with respect to the medial axis sampling condition is valid with respect to our sampling condition.*

Proof. Choose $\epsilon = \Theta(\frac{1}{\beta})$.

3 The algorithm

The main idea of our algorithm is that we first detect the edges that can be justified as being 'smooth'. Then starting from these edges we explore potential corner areas, possibly removing some of the edges previously discovered as smooth.

The following is a high-level description of our algorithm:

1. Compute the Delaunay triangulation of S
2. Among the Delaunay edges determine the set of all edges e which have two balls of radius $\beta \cdot |e|/2$ empty of other samples. Color these edges *red*.
3. Uncolor all red edges that belong to red chains of length less than 3.
4. Let T be the set of samples which are adjacent to exactly *one* red edge
5. As long as there are elements in T, remove one of them and start exploration of a potential corner. If successful, add this corner to the set M of detected corners.
6. Remove interfering corners from M
7. Remove some red edges that interfere with the corners in M to get the final polygonal reconstruction $H(S)$.
8. Construct a collection of smooth curves Γ' from $H(S)$ by adding small 'caps' and corners points

Of course, the most interesting part is how to actually explore potential corners. From now on, we assume that a corner is represented as two sequences corresponding to the upper and lower leg ending in that corner.

3.1 Exploring a corner The idea of the corner exploration is that we consider a sample $s \in T$ as starting point of a potential corner. s is adjacent to exactly one red edge e_s, and we first try to continue this red edge with a blue edge. As we do not know the orientation of e_s, i.e. we do not know where the 'outside' of the potential corner is, we simply try both possibilities.

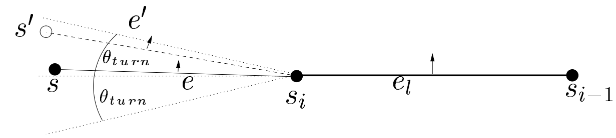

Figure 7: Picking the next edge of the potential corner

For this step and the steps to follow, a crucial part is how to determine the next edge in a leg. This is done using the following procedure (also look at Figure 7):

FindNextEdge(s_i, s_{i-1}): Let $e_l = (s_i, s_{i-1})$ be the last edge detected in one leg of a corner. Assume we are also given an orientation, i.e. we know where the 'outside' of the corner is (which implies that there are empty β- and β_{low}-balls on that side).

Let M be the set of all blue or red edges $e' = (s', s_i)$ which make a turn of less than θ_{turn} with e_l and which have empty β_{low}- and β-balls on the same side as e_l.

As next edge $e = (s, s_i)$, pick the shortest among the edges in M.

Having found this adjacent blue edge e_b, we use the fact that this blue edge builds a Delaunay triangle with a sample o on the opposite leg (follows from the sampling condition). If o has no adjacent red edge, we abort the exploration, otherwise we pick one of the (potentially two) red edges adjacent to o – let's call it e_o which does not contain a sample already part of e_s or e_b. Then we 'walk' along both (potential corner) legs until finding the corresponding corner. As we do not know in advance neither the orientation of e_o, i.e. where with respect to e_o the 'inside' of the corner is, nor the direction of e_o, i.e. in which direction the corner lies, we just try all four possibilities [4]. Similarly, there might be cases, where only one of the (potentially two) red edges adjacent to o leads to a correct corner exploration, so we also try both possibilities[5].

Following the legs actually works step by step. We first determine for each of the two current reconstructed legs whether there exist potential continuation edges (using the **FindNextEdge** procedure). We take the shorter of them and then verify whether the things reconstructed so far justify a corner. If so, we store it in M but nevertheless continue to follow the candidate legs until either

- for both sequences no continuations are found, or

[4] In many cases it is possible to exclude some of the directions and orientations, but there are examples where it is not possible to decide which direction/orientation is the right one.

[5] There is one case, where one of the red edges leads to an incorrect corner exploration, namely when one of them is a corner spanning edge.

- a sample is picked a second time, or
- the slope between any two segments in one candidate leg is larger than θ_{slope}
- the supporting lines of two edges in different legs intersect *both* in opposite direction to the corner w.r.t. e_1 and e_2.

When this procedure terminates, we take the last corner (which also means the 'largest') that was successfully verified (if any) and add it to the set M of potential corners (with all edges of the grown corner ball as determined in the following verification stage).

3.1.1 Verifying a corner

The task of the verification stage, which is called after every continuation step, is to check whether the connections found so far make up a justifiable corner. To do this, we compute a tentative corner point and check whether there exists a large enough ball to cover all of the blue edges in our current reconstruction which is empty of other samples (samples that are not part of the corner to be verified).

Let s_u and s_l be the last samples in the upper and lower legs reconstructed so far, and let e_u and e_l be the corresponding last edges. To determine a tentative corner, we consider the two cones of angle $2 \cdot \theta_{turn}$ at s_u and s_l w.r.t. e_u and e_l. The tentative corner point C is determined as (in this order):

- if s_u is contained in the other cone, set $C = s_u$ (and vice versa)
- if the cones do not intersect, the corner verification has failed
- if the cones intersect, take the 'inner' intersection point of their borders.

We then determine the maximum distance D from C to any point in one of the sequences found so far which has a blue adjacent edge. We extend the two candidate legs by red edges (if such exist) as long as the total turn (including the edges from the tentative corner point C to the last samples of the candidate legs) is less than $\theta_{slope} + \theta_{turn}$ in each leg and as long as the furthest point of each leg has distance less than $f_{grow} \cdot D$ from C. If no such red edges can be found, the verification fails, otherwise, we consider the ball around C with radius $f_{grow} \cdot D$. If it contains other samples than the one present in the two sequences, the verification fails, otherwise we check that all blue edges only contain in their inner β_{low}-balls samples of the opposite leg within the grown ball, and that any connection between samples of different legs intersects the interior of the corner.

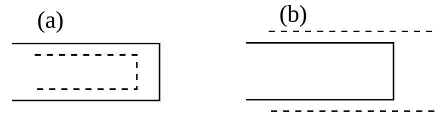

Figure 8: Cases of agreeing overlap

3.2 Removal of interfering corners

Our algorithm by now has produced a collection of possible corners (represented as the two reconstructed legs ending in that corner) which might possibly interfere with each other, where we say that two corners interfere with each other, if the overlay of the corresponding graphs has a degree 3 vertex. We distinguish two kinds of interference: *overlap* and *intersection*. Two corners

- *overlap* If the degree 3 vertices are only caused by (at most two) corner spanning edges which cross the interior of the other corner. We also distinguish between:

  agreeing overlap: if both corners point into the same direction, see Figure 8 for a schematic outline of these cases.

  disagreeing overlap: if the corners do not point into the same direction. This case cannot happen due to the the sampling condition where we disallow that the supporting lines of two edges on different legs intersect in opposite direction to the corner for *both* edges.

- *intersect* if they interfere but do not overlap

We first get rid of the intersecting corners by just deleting any pair of corners that intersects each other. For the remaining overlapping ones, we always delete the corner that starts inside the other one, so in Figure 8, the dashed corners would be deleted.

3.3 Removal of interfering red edges

At that stage we have identified a set of potential corners, but some of them might interfere with red edges found in the first step, i.e. they might touch or cross a potential corner causing a degree 3 vertex. We will prove later on that these red edges cannot be part of the correct reconstruction, so we simply delete them.

4 Correctness of the algorithm

First we have to prove is that if a collection of curves Γ is correctly sampled according to our sampling condition, then the correct reconstruction is part of the output graph $H(S)$ of our algorithm ('Good edges are captured'). Of course we could satisfy this by returning the complete Delaunay triangulation of the sample set. So as in [5] we also have to prove that there exists a collection of curves Γ' for which S is a valid sampling (with a slightly weaker sampling condition) and $H(S)$ is the correct reconstruction of S with respect to Γ' ('Captured edges are good'). So Γ' is in some sense a certificate for the reasonability of each edge our algorithm has constructed.

When choosing appropriate parameters for the sampling condition and the algorithm like

$$\begin{array}{ll} \beta = 2 \Leftrightarrow \theta_{beta} = 30^0 & \beta_{low} = \frac{2}{\sqrt{3}} \Leftrightarrow \theta_{low} \approx 60 \\ \theta_{turn} = 10^o & \theta_{slope} = 30^o \\ \theta_{ball} = 30^o & f_{dia} \approx 2.84 \\ f_{shrink} \approx 4.71 & f_{grow} \approx 1.86 \end{array}$$

we obtain the following two theorems:

THEOREM 4.1. *Let Γ be a collection of open or closed curves possibly with corners, S be a set of samples from that collection meeting our sampling condition. Then each edge of the correct reconstruction $G(S,\Gamma)$ is present in the graph H returned by our algorithm with the only possible exception of edges spanning a corner of the curve.*

THEOREM 4.2. *For any input sample set S, our algorithm returns a graph H and a collection of curves Γ' such that S is a valid sample set for Γ' with:*

$$\begin{array}{ll} \beta' = \beta & \beta'_{low} = \beta_{low} \\ \theta'_{turn} = \theta_{turn} & \theta'_{slope} = \theta_{slope} + \theta_{turn} \\ f'_{dia} = \frac{1}{2} & \theta'_{ball} = \theta_{ball} \\ f'_{shrink} = \frac{f_{grow}}{\sqrt{1-\frac{1}{\beta^2}}} & \end{array}$$

and $H = H(S,\Gamma')$, i.e. H is the correct reconstruction of S with respect to Γ'.

Remarks One might wonder why Theorem 4.1 excludes corner spanning edges of $G(S,\Gamma)$ in the guarantee for the output of our algorithm. One reason for that is that if the two legs of a corner are sampled so densely that there is no blue edge, our algorithm cannot find a starting point for the corner exploration, so it finds all edges except for the corner spanning edge (see Figure 9). Observe that this also makes sense, because if the two legs are very densely sampled, it may well be the case that the original curve does not have a corner there but just two endpoints. We will need this 'conservative' behaviour of our algorithm later on when we modify the sampling condition and our algorithm to get a result of the type: *For every collection of curves with corners and endpoints, there exists a sampling such that our algorithm outputs exactly the correct reconstruction*.

The other reason for a corner spanning edge possibly not being detected is that our algorithm manages to extend the corner further by one or more samples. See Figure 10. Here the actual corner would be the dashed one, but the algorithm was able to justify the extended corner which also includes

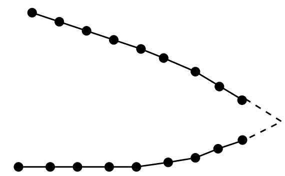

Figure 9: Densely sampled legs of a potential corner

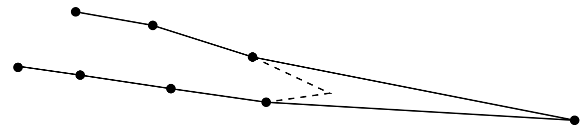

Figure 10: Corner extended by another sample

the sample s (which has to be outside the unshrunken corner ball of the correct corner, of course). Note that this is mainly due to our relaxed sampling condition which uses a constant for θ_{turn} independent of the angle at the actual corner. Later we will show how to exclude that case as well without sacrificing the constant θ_{turn} angle.

In the following we elaborate on these two theorems but do not include the proofs which can be found in the full version of this paper.

4.1 Good edges are captured (Theorem 4.1) Let $G(S,\Gamma)$ be the correct reconstruction of S with respect to a collection of curves Γ. We will show in the following that if S is a valid sampling of Γ, then every edge of of $G(S,\Gamma)$ will be detected and 'survive', and therefore be present in the output H of our algorithm (with the exception of corner spanning edges).

The following lemma does not require proof:

LEMMA 4.1. *Smooth edges are detected and colored red by the algorithm after the first 2 steps.*

What we have to prove now is that smooth edges will not be killed later on because of interference with a potential corner:

LEMMA 4.2. *A smooth edge cannot:*

1. *'touch' a (wrong) corner from outside at a sample which is not a corner sample*
2. *'touch' a corner at a corner sample*
3. *'cross' an incorrectly detected corner*

So now we know that every smooth edge of the correct reconstruction will survive the stages of our algorithm and hence be present in the output. Let us now consider the non-smooth edges of one particular corner. First we will show that there is a canonical element $s \in S$ such that if the algorithm starts a corner exploration from s, it will detect a potential corner which covers all the edges of the real corner we are considering (possibly even more, as we have mentioned before).

To prove this we first have to state a small lemma which implies that if we are given a correct part of the sequence of samples on either leg (together with an orientation where the 'outside' is), our procedure `FindNextEdge()` will find the next edge of this leg (if it exists).

LEMMA 4.3. *Let $e = (s_1, s_0)$ be an edge of the correct reconstruction within a corner area, oriented such that the 'outside' of the corner lies to the left of $\overrightarrow{s_1 s_0}$, and let s be the other sample which s_1 is adjacent to. Assuming that s lies on the same leg as s_1 and s_0, then there is no other sample s' such that the following conditions all hold at the same time:*

- *the turn-angle between e and $e' = (s', s_1)$ is less than θ_{turn}*
- $d(s_1, s') \leq d(s_1, s)$
- *there is an empty β-ball to the left of $\overrightarrow{s', s_1}$*

Now we know that if we have somehow managed to find the right 'start' of the corner, i.e. correctly determined the start of the two legs ending in that corner, there will be a time during the algorithm's execution where exactly the correct edges of the corner have been detected (except for the corner spanning edge). This can be easily seen by the fact that we always extend with the shorter continuation edge and therefore first all edges within the corner area are picked before connecting to outside the corner area.

To show that for every corner there exists a good starting point for the exploration, follow the red edges of the correct reconstruction on both legs until hitting an edge of the correct reconstruction which is not red (or reaching a corner sample). Let p_l, p_u be the samples obtained by this procedure. If both are corner samples, we don't even have to start a corner exploration as all edges of the correct reconstruction are already detected (except for the corner spanning edge). Otherwise at least one of them is not a corner sample and has no other (wrong) adjacent red edge. If it is exactly one, then this is a good starting point for the corner exploration, if both are not corner samples and do not have other (wrong) adjacent red edges, one of them will be a good starting point. We call this the *canonical corner exploration*.

It remains to prove that at the point when exactly the correct edges have been detected, the verification test will pass.

LEMMA 4.4. *At the time when the correct edges of a 'real' corner have been detected, the verification test will pass for $f_{grow} \leq (f_{shrink} - 1)/2$.*

So now we know that for each corner of the correct reconstruction, there is at least one reconstruction detected by the algorithm which covers all edges of that corner. It remains to show that for every corner, one of them survives the next stages.

LEMMA 4.5. *The reconstruction of a correct corner cannot be intersected by the reconstruction of a 'wrong' corner.*

LEMMA 4.6. *At least one corner exploration covering (at least) all edges of a corner in the correct reconstruction survives all stages of the algorithm.*

We have proven that every edge of the correct reconstruction is detected by the algorithm and survives till the end (expect for corner spanning edges). So 'all good edges are captured'.

4.2 Captured edges are good (Theorem 4.2)

Basically almost all statements of Theorem 4.2 follow directly from the algorithm. The only statement that requires a proof is the statement that for every corner, the ball of radius $r \cdot \sqrt{1 - \frac{1}{\beta^2}}$ does not intersect any segments of the output graph which are not part of the the two legs ending in that corner, where r is the radius of the extended corner ball. But this can be easily seen, as the distance of any edge e to the center of the extended corner ball must be greater than $r \cdot \sqrt{1 - \frac{1}{\beta^2}}$, since this e must have an empty β-ball on that side, hence the ball of radius $r \cdot \sqrt{1 - \frac{1}{\beta^2}}$ cannot intersect any of these segments.

4.2.1 Construction of a collection of Witness Curves Γ'

As our sampling condition does not directly refer to the curvature of the curve, we can construct witness curves by simply taking the polygonal reconstruction computed by our algorithm and adding very small 'caps' at every sample which is adjacent to two non-corner spanning edges. Corner spanning edges are replaced by two edges to the corner point estimated by the algorithm.

4.2.2 Witness Curves with respect to the medial axis

Using the same idea as in [5], we can prune the output of our algorithm even further and then construct curves as in [5], which are then witness for the sampling condition with respect to the medial axis.

5 How to obtain *exactly* the correct reconstruction ?

The algorithm as outlined so far guarantees that all edges of the correct reconstruction G of the original collection of curves Γ are present in the output H of our algorithm (with the exception of the corner spanning edges), but there might be additional edges in the output, though our algorithm can 'justify' each of them.

The ultimate goal, of course, is to find a sampling condition and an algorithm which for any collection Γ of curves

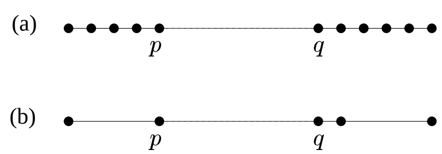

Figure 11: Connect p and q or not ?

with endpoints and corners guarantees that the output of the algorithm is exactly the correct reconstruction $G(S, \Gamma)$, if the sample set S conforms to this sampling condition.

To reach this goal, one first has to restrict the possible curves allowed in Γ. Given any sample set S, it is always possible that the original collection of curves Γ consists exactly of those sample points, i.e. each curve in Γ is degenerate in a sense that it only consists of a single point. There is no way the algorithm can detect this. So it seems reasonable to restrict the curves allowed in Γ to be non-degenerate, i.e. each curve must consist of more than just one point.

The second difficulty arises in cases as sketched in Figure 11. With the current sampling condition and algorithm, the algorithm would connect p and q even if they are endpoints in the correct reconstruction, provided there are no samples in the neighbourhood which are inside the β-balls of (p, q).

Note that in Figure 11,(a) it really seems reasonable *not* to connect p and q as the edge (p, q) would be very long compared to the other edges in that chain. On the other hand, one should definitely not reject an edge (p, q) just because let's say q's other adjacent edge is very short as shown in Figure 11,(b).

So we propose the following oversampling condition:

Oversampling-Condition: Let $e = (p, q)$ be an edge in a potential reconstruction of a sample set. Furthermore let $q_1, q_2, q_3, \ldots q_k$ be the samples in the chain when following the chain adjacent to q, and let p_1, p_2, p_3, $\ldots q_k$ the samples adjacent to p.

We say e is not reasonable due to oversampling, if either

- $q_1, \ldots q_k$ exist and $\forall i = 1 \ldots (k-1)$ we have $|e| > f_{stretch} \cdot d(q_i, q_{i-1})$,
- $p_1, \ldots p_k$ exist and $\forall i = 1 \ldots (k-1)$ we have $|e| > f_{stretch} \cdot d(p_i, p_{i-1})$,

If we use for example $k = 5$ and $f_{stretch} = 2$ this means that for one particular edge e, the longest edge amongst its 5 neighbours on either side must be at least half as long as e.

So we add the following condition:

Minimum Sampling Condition: For each component of the collection of curves Γ, there must be either 0 samples or more than k samples.

It should be clear how to modify the algorithm to reject edges which are unreasonable due to oversampling. The way we can now find an appropriate sample set to get rid of the gap closing edges is that we first take a valid sample set with respect to our original sampling condition and then add samples on the edges close to a gap until this edge gets rejected due to the oversampling condition. We can also turn this directly in a sampling condition:

Gap Marking: For any edge e of the correct reconstruction, all diametral balls around e with radius $\leq |e|/2 + |e| \cdot (k-1) + f_{stretch} \cdot |e|$ must not contain the endpoint of another component of the curve than the one which e belongs to in the intersection with the ball.

LEMMA 5.1. *If the gap marking condition is fulfilled for all edges of the correct reconstruction, no endpoints of the correct reconstruction are closed by our algorithm.*

Another problem is the fact that our algorithm might not close some corners, if the samples on both legs are so dense that there is no blue edge which could trigger a corner exploration (see Figure 9). We can circumvent this by postulating:

Corner Triggering: For any corner there must be a blue edge in the correct reconstruction.

Observe that it is very simple to generate such a blue edge by placing a sample close to the actual corner point of the corner. Either both edges connecting the formerly last samples of either leg to this additional sample are red then we are done anyway, or one of them is blue, so it triggers the corner exploration.

Furthermore our algorithm might extend a corner further than it is supposed to do as we have seen in Figure 10. This can be either avoided by enforcing the oversampling condition or again by placing a sample point close to the actual corner point which is then chosen by both legs and hence terminates the corner exploration.

So with this additional sampling conditions and a slightly restricted definition of curves, we obtain the following:

THEOREM 5.1. *For every collection of curves Γ there exists a finite sample set such that our algorithm exactly returns the correct reconstruction.*

6 Running Time

We haven't attempted to optimize the running time of our algorithm. Assuming that of the $n = n_s + n_c$ samples we have n_c samples in corner regions and n_s samples in regular parts, we can obtain a running time of $\mathrm{O}(n_c \cdot n^2)$ with a very naïve implementation. In practice, it seems to be dominated by the time to compute the Delaunay triangulation.

7 Implementation and Experimental Results

We have implemented our algorithm (currently without the oversampling condition) which seems to behave well in practice, even when using the theoretical parameters. See Figures 12, 13, 14 for the output of the CRUST ([2]), the CONSERVATIVE-CRUST ([5]) and our algorithm for one particular sample set. We have not yet performed extensive experimentation. In particular, we have not compared our algorithm to that in [6].

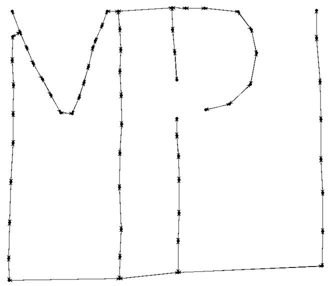

Figure 12: The Crust algorithm

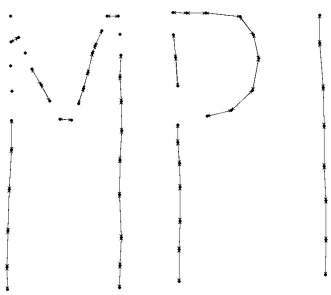

Figure 13: The Dey-Mehlhorn-Ramos algorithm

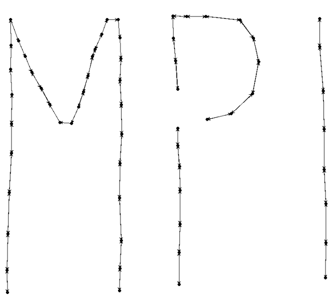

Figure 14: Our algorithm

8 Concluding Remarks

We presented an algorithm for curve reconstruction which can provably handle a collection of curves with corners and endpoints and also introduced a new sampling condition which is not as restrictive as the sampling conditions based on the medial axis. Even in practice the algorithm seems to perform well. We also proved that for any collection of curves with corners and endpoints, there exists a finite sample set from that collection for which a slight modified version of our algorithm outputs exactly the correct reconstruction.

In two dimensions, the problem of reconstructing open and closed curves with branching points is still open. In three dimensions, algorithms with a guarantee only exist for closed smooth manifolds. No algorithms have been found yet that can handle sharp corners and ridges well, neither in theory nor in practice. The main idea of our algorithm, first to detect parts of the curve that are likely to be smooth and then explore potential corners might also work in three dimensions.

References

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

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

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

[4] T. K. Dey and P. Kumar. A simple provable algorithm for curve reconstruction. *Proc. 10th. ACM-SIAM Sympos. Discr. Algorithms*, (1999), 893-894.

[5] T. K. Dey, K. Mehlhorn and E. A. Ramos. Curve reconstruction: Connecting dots with good reason. *Computational Geometry Theory and Applications*, 15 (1999), 229–244.

[6] T. K. Dey and R. Wenger. Reconstructing curves with sharp corners. *Proc. 15th Ann. ACM Sympos. Comput. Geom.*, (2000), 233–241.

[7] T. K. Dey and R. Wenger. Guaranteed Reconstruction of Curves with Sharp Corners. *Manuscript*.

[8] H. Edelsbrunner. Shape reconstruction with Delaunay complex. *LNCS 1380, LATIN'98: Theoretical Informatics*, (1998), 119–132.

[9] H. Edelsbrunner, D. G. Kirkpatrick, and R. Seidel. On the shape of a set of points in the plane. *IEEE Transactions on Information Theory*, vol. 29 (4) (1983), 71–78.

[10] L. H. Figueiredo and J. M. Gomes. Computational morphology of curves. *Visual Computer*, vol. 11 (1994), 105–112.

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

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

[13] D. G. Kirkpatrick and J. D. Radke. A framework for computational morphology. In "Computational Geometry", edited by G. Toussaint. North Holland, 217–248.