Volume xx (200y), Number z, pp. 1–18

An Efficient Algorithm for Determining an Aesthetic Shape Connecting Unorganised 2D Points

S. Ohrhallinger[1,2] and S. Mudur[1]

[1]Concordia University, Montréal, Canada, [2]Vienna University of Technology, Austria

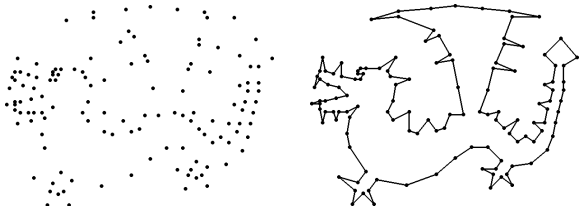
Abstract

We present an efficient algorithm for determining an aesthetically pleasing shape boundary connecting all the points in a given unorganised set of 2D points, with no other information than point coordinates. By posing shape construction as a minimisation problem which follows the Gestalt laws, our desired shape B_{min} is non-intersecting, interpolates all points and minimises a criterion related to these laws. The basis for our algorithm is an initial graph, an extension of the Euclidean minimum spanning tree but with no leaf nodes, called as the minimum boundary complex BC_{min}. BC_{min} and B_{min} can be expressed similarly by parametrising a topological constraint. A close approximation of BC_{min}, termed BC_0 can be computed fast using a greedy algorithm. BC_0 is then transformed into a closed interpolating boundary B_{out} in two steps to satisfy B_{min}'s topological and minimization requirements. Computing B_{min} exactly is an NP-hard problem, whereas B_{out} is computed in linearithmic time. We present many examples showing considerable improvement over previous techniques, especially for shapes with sharp corners. Source code is available online.

Categories and Subject Descriptors (according to ACM CCS): I.3.3 [Computer Graphics]: Picture/Image Generation—Line and curve generation I.3.5 [Computer Graphics]: Computational Geometry and Object Modeling—Boundary representations I.4.8 [Computer Graphics]: Scene Analysis—Surface Fitting

1. Introduction

The "connecting the dots" problem using only point coordinates to find a closed, interpolating and non-intersecting curve in 2D is known to be very difficult. We constrain this general problem to point sets which are not randomly distributed or extremely non-uniformly spaced, i.e, point sets in which human observers can "see" the desired shape.



(a) Point set (b) Shape boundary

Figure 1: a) Unorganised point set P. b) A pleasing shape connecting P, which is a closed, non-intersecting and interpolating curve.

As in [OM11] our approach derives its requirement of a closed non-intersecting curve from the Gestalt law of *Closure* and minimises globally a property related to both the laws of *Proximity* and *Good continuity* to define the desired interpolating shape denoted as B_{min}.

The problem of finding the interpolating polygon which minimises a metric, globally, is known to be NP (Non Polynomial)-hard. Since even humans cannot uniquely identify a shape in extreme point distributions, it is only reasonable to expect that the algorithm also does not attempt to solve this problem for point sets with highly non-uniform spacing or extremely sparse distribution in 2D space. Further, B_{min} for such point sets is not robust with respect to minor point displacements, so we exclude such extreme point distributions.

The main contribution of our algorithm is to provide an efficient solution, which improves significantly on previous

work for reconstructing closed curves interpolating sparsely spaced points with sharp corners in the boundary shape (see Figure 1).

An important claim is the linearithmic computational complexity of our algorithm, given that we approximate an NP-hard problem. The computational complexity is contained by defining intuitive transforming steps that start from a simple extension of the Euclidean Minimum Spanning Tree ($EMST$) and lead us to find a shape boundary, B_{out} which interpolates all points while trying to minimise the defined metric. The starting graph is a greedy approximation of the $EMST$ extension which we have called as minimum boundary complex BC_{min}. Our other contribution is the way we relate $EMST$, BC_{min} and B_{min} as minimum spanning graphs differing only in topological constraints. We believe that it is this which enables our method to find a closed shape boundary from much sparser point sets than previous work. Since, the transforming steps mentioned above basically try to satisfy B_{min}'s topological constraint and at the same time minimise the defined metric, we conjecture that resulting B_{out}, though it is not B_{min}, is quite a close approximation. We show many examples with different kinds of point data sets to illustrate this.

2. Background Definitions

- $\mathbb{R}^2$ is two-dimensional Euclidean space.
- $||n||$ denotes the Euclidean norm in $\mathbb{R}^2$ for a vector n.
- P in $\mathbb{R}^2$ denotes the given unorganised *Point set* P for which an aesthetic closed, non-intersecting and interpolating piece-wise linear curve has to be constructed.
- $DG(P)$ is the *Delaunay graph* of the point set P with all its elements in $\mathbb{R}^2$. The letters v, e and t denote vertices, edges and triangles, respectively. We shall use G to denote any subset of DG.
- *Euclidean minimum spanning tree* $EMST$ is the tree spanning all points in P such that the sum of its edge lengths is the minimum. $EMST \subset DG$ (Jaromczyk and Toussaint [JT92]).
- B is a non-intersecting and closed piece-wise linear curve interpolating all the points of P. The *minimum boundary* B_{min} is the B with minimum boundary length. Finding B_{min} exactly is NP-hard.
- *Enclosing boundary* B_e is the sub-set of edges of a spanning graph $G \subseteq DG(P)$ which can be traversed in a fixed orientation such that edges of G which are not part of B_e are all contained on one side of B_e, denoted as its *inside*. We denote the set of triangles contained at the inside of B_e as its *inside triangulation*. The orientation is assigned such that every triangle that is inside B_e is also inside the convex hull of P.
- *Manifold* vertex v_i in B_e is visited exactly once when traversing B_e in a given orientation. A B_e with only manifold vertices is itself *manifold*.

3. Related Work

In the literature we find two major approaches. One is to cast this problem as 2D shape reconstruction by considering the points as samples on a known 2D curve(s). This then makes it possible for algorithms to work for point sets satisfying specified sampling criteria. Usually these criteria impose quite strict conditions with regard to point spacing properties, requiring high density, uniformity and smoothness. The second approach is to view this problem as a global search through all possible solutions. Below, we briefly review previous work using these two approaches.

3.1. Local Sampling Condition Approach

Algorithms based on the sampling-oriented approach connect the points using edges in the Delaunay Graph (DG) and results have shown that this is a very reasonable choice. The DG has the property of maximizing its angles and minimizing its edge lengths, which conform to the Gestalt laws of good continuity and proximity.

α-shapes [BB97, EKS83], Figueiredo and Gomes [FMG94], β-skeleton [KR85], γ-neighbourhood graph [Vel93] and r-regular shapes [Att97] are among the early methods which worked only on smooth and uniformly sampled point sets. For example, α-shapes requires user-specification of a global constant which depends on sampling. It does not work for non-uniformly sampled point sets.

Amenta et al. [ABE98] with their *Crust* algorithm introduced the concept of local feature size which allows reconstruction from non-uniformly sampled point sets with a minimum angle α between edges of the reconstructed piece-wise boundary. The minimum angle α is derived from their ε-sampling condition. The stated sampling requirements of the *Crust* method ($\varepsilon < 0.252$, requires $\alpha > 151.05^\circ$) and its successors [DK99, DMR99] ($\varepsilon < \frac{1}{3}$, requires $\alpha > 141.62^\circ$) are however quite restrictive in theory and difficult to ensure in practice. The *Gathan* algorithm from Dey et al. [DW01] handles sharp corners and in its extension to *GathanG* proves a combined sampling condition for smooth and corner parts of the curve [DW02]. It does not take aesthetic aspects into consideration. In spite of this, it provides in our opinion, to date, the best sampling-oriented solution for this problem of 2D shape reconstruction from sparse point sets with sharp corners. And this is also evident from the comparison in [ZNYL08].

Zeng et al. use the two properties of proximity and smoothness derived from Gestalt laws but still require rather dense sampling in sharp corners [ZNYL08]. Some improvements on these aspects have been made in [NZ08], but they rely very much on user-tuned parameters.

The output produced by algorithms using a local sampling criterion is a manifold which can be multiply connected and

(a) Connected (b) Disconnected

Figure 2: Point set of two far circles. a) Requiring global connectedness. b) Only requiring (local) closure.

possibly bounded. A specific requirement, say a manifold which is closed and singly connected, can only be guaranteed for sufficiently dense sampling, imposing very restrictive criteria. Otherwise, the results are not predictable. This can be seen later for example in Figure 13, which shows a number of such cases. Of course, local sampling conditions are designed specifically to make no assumptions on the shape, but rather how the shape was sampled. In comparison, we have posed this problem as one requiring a shape with a closed and singly connected manifold boundary. Further, our observation from the many experiments conducted is that enforcing the Gestalt law of *Closure* actually yields more pleasing shapes. And if one indeed desires to get an open shape, then an openness condition, such as large distance between points, very sharp turns or others, can be applied to the closed curve to make it open.

Also, if multiple connected components are required, then these can be produced for a point set by dropping the global condition of spanning tree enforcement and extracting the interpolating manifold from each disjoint set separately. This prefers *Proximity* while still fulfilling *Closure* as a local property. The choice is then up to the user (see Figure 2). However, we focus on the generic problem of producing a single closed manifold shape.

3.2. Global Search Approach

A first attempt using a global search approach is the one presented in [GB97]. They construct spanning Voronoi trees and select the one with minimal length by integer programming, with $O(n^2 \log n)$ complexity. It does not work well for sharp angles and non-uniform sampling; obviously it prunes good solutions too early.

Giesen shows in [Gie99] that the solution for the Euclidean travelling salesman problem (*ETSP*), called a *tour*, can reconstruct the shape for sufficiently dense sampling. He presents two algorithms, but no results, and only an existence guarantee. Based on that, [AM00] show that such a tour reconstructs shapes also for non-uniform sampling and that it solves the NP-hard problem in polynomial time, if constrained with the sampling condition as presented in [ABE98]. But the proof given is only for an extremely restrictive $\varepsilon < \frac{1}{20}$, which requires $\alpha > 174.27°$. For unrestricted sets Arora [Aro96] gives a $(1+1/c)$-approximation to the optimal ETSP tour in $O(n(\log n)^{O(c)})$. These approximations are however not optimal and our experiments showed that non-optimal solutions are often poor from the aesthetic shape perspecive.

In [AMNS00] the exact TSP based solution is compared with *Crust*-type family of algorithms and six other TSP approximation solutions based on different heuristics. They note that these TSP-heuristics all fail for certain curves containing sparse sampling which the exact TSP method handles well. They also mention that the exponential complexity of the TSP decreases with denser sampling. With the exception of [Gie99], these methods do not require user-specified parameters. Unfortunately, finding the exact solution using a naive TSP solver takes unreasonable time $O(2^n)$ even for small P. The *concorde* exact TSP solver [ABCC11] scales sub-exponentially and can take hundreds of CPU-years for medium-sized point sets. Its complexity is discussed in detail here [Hoo09]. While, in principle, a TSP solution constrained to DG would yield B_{min}, we consider the TSP as too generic to be applicable for the problem we have stated. Our focus is on an algorithm for efficient shape construction to produce aesthetic shapes as seen by humans and it only needs to work on point sets that are reasonable in their spacing.

In [OM11] Ohrhallinger and Mudur show that for a certain class of point sets there exists a relation between minimum perimeter polygon in DG and the Euclidean minimum spanning tree ($EMST$) of P. This relation is characterised by well-defined edge exchange operations. While their algorithm gives very good results for sharp corners, it cannot guarantee linearithmic complexity since in some cases a global search of the solution space may be required. Their main contribution is the approach to formulate curve reconstruction as a minimisation problem, by relating to properties of the Gestalt laws for aesthetic shape.

The algorithm presented in this paper on the other hand is very different, even though it also minimises the same criterion of length. In 2D, it turns out that length minimisation relates very well to the above-mentioned Gestalt laws. We base our algorithm on the observation that for point sets, except in those which are random or extremely non-uniformly spaced, the $EMST$ graph characterises the boundary shape rather well. However, there are leaf vertices in $EMST$ and there should be no leaf vertices in the interpolating, closed manifold curve.

Based on this observation we formulate an extension to the $EMST$ by requiring that each vertex must have at least two incident edges. Our experiments show that the enclosing boundary of the resulting graph approximates much better the shape boundary (see Figure 3c). This is so because the extended graph shares a large sub-set of edges with B_{min}, since it has the same minimization objective and differs only in the topological constraint, which is slightly different. It is not a tree. We name it the *minimum boundary complex* (BC_{min}).

(a) BC_0 (b) BC_{min}

Figure 4: a) BC_0 reconstructed from a point set P by our greedy Algorithm 1. b) $BC_{min}(P)$, which for this case is also equal to $B_{min}(P)$. BC_0 is transformed into B_{min} by our algorithm as described later on.

Computation of BC_{min} and B_{min} is NP-hard. However, given the relaxed topological constraint of greater than two incident edges per vertex, we are able to use a greedy algorithm to compute an approximation to BC_{min}, which we call as BC_0. To this BC_0, we then apply two steps derived from the requirements of B_{min}. The first is that we must have exactly two edges per vertex and the second is minimisation of length. The output we get, which we shall denote as B_{out}, is a closed, non-intersecting, manifold boundary. Experiments show that the results we get are much better than previous methods, especially for sparser point sets. We also prove that like other Delaunay-based methods based on a local-sampling condition, the complexity of our method is $O(n \log n)$, which is a major advantage over [OM11].

4. The Minimum Boundary Complex BC_{min}

Definition 1 Let $BC = (V, E)$ be a connected graph with $E \subset DG(P)$ spanning the point set P such that each vertex $v_i \in V$ has ≥ 2 incident edges $e_i \in E$. The criterion minimised is $len(BC) = \sum_{e_i}^{E} BC$. Then the minimum BC must satisfy:

$$BC_{min}(P) = \arg\min(len(BC_i), BC_i \subset DG(P)) \qquad (1)$$

An approximation of BC_{min} can be constructed using a greedy algorithm in $O(n \log n)$ time, which we denote as BC_0 (see Algorithm 1). BC_0 is used subsequently as the starting graph and it may not be the same as BC_{min} due to the greedy algorithm terminating at a local minimum. In the cases where $BC_0 \neq BC_{min}$ (see Figure 4), it does not matter much since the post-processing steps anyway transform it into another graph with topological constraints and minimization objective, the same as for B_{min}.

Lemma 1 Given a point set P with n points and its Delaunay graph $DG(P)$, BC_0 can be constructed in $O(n \log n)$ time.

Proof Creating PQ inserts at most the $O(n)$ edges of DG into a sorted list, therefore complexity is $O(n \log n)$. The while loop is executed at most for each edge in DG, $O(n)$ times. Testing for and keeping track of connectedness is done via a disjoint set. Its operations are an amortised $O(\alpha(n))$, where $\alpha(n)$ is the inverse of the Ackerman function (see Theorem 5 in [FS89]). The entire algorithm is therefore $O(n \log n)$. □

Input: P, DG
Output: BC_0
$BC_0 = \{\}$;
Insert all edges $e_i \in DG$ into a priority-queue PQ sorted by ascending $\|e_i\|$;
while BC_0 *is not a connected component or a vertex* $v_i \in P$ *has* < 2 *incident edges in* BC_0 **do**
  Remove first edge e_i from PQ;
  if (e_i *connects two connected components in* BC_0) $\vee$ (e_i *is incident to a leaf vertex in* BC_0) **then**
    Insert e_i into BC_0;
  end
end

Algorithm 1: Construction of BC_0

The algorithm for computing BC_0 relates directly to the Gestalt law of *Proximity*. *Closure* is fulfilled as well by the constraint of minimum two incident edges for every interpolated point. *Good continuity* is not always strictly followed, since it can conflict with closure (for an example see Figure 1 where the uppermost point of the tail and the rightmost point of the wing are not connected as one would expect following good continuity). However it is fulfilled implicitly, since restricting the edges to DG maximises angles between edges and selects small edges, which in turn correlates well to low curvature.

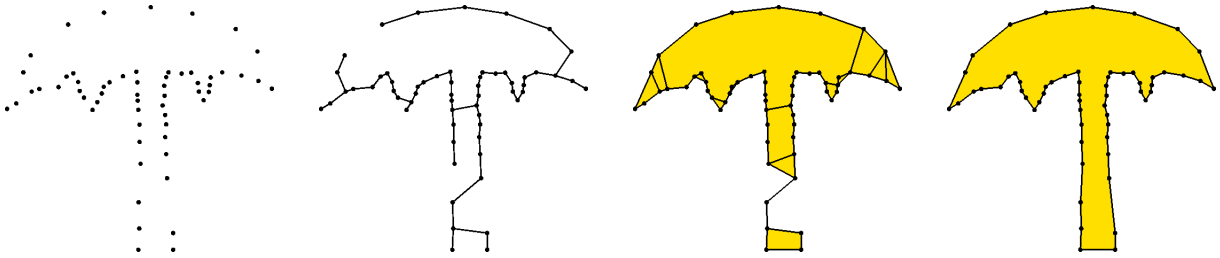
Algorithm 1 yields global closure by requiring connectedness. If we instead aim for local closure (as shown in Figure 2b), only the conditions requiring connectedness have to be dropped from the algorithm. This does not affect its time complexity. Note that the algorithm may not have a unique result if the DG contains edges of equal length. However, this is not a problem, as based on perception either result will be equally valid.

5. The Main Algorithm

5.1. Overview

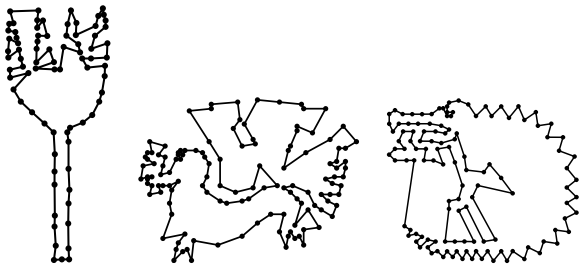
One major difference between BC_0 and B_{min} is that BC_0 has non-manifold vertices (degree > 2) (see Figure 3c; for example the three vertices in the lower part of the stem).

Hence the goal of subsequent steps of our method is to transform BC_0 so that its vertices are each contained in exactly 2 edges to yield B_{out}. For this, we start with B_e, the enclosing boundary of BC_0 and transform it. Let us note that B_e has both manifold and non-manifold vertices, and further may have vertices in its interior. To better understand the steps in this transformation process, we employ for B_e the metaphor of a yet partially inflated air mattress. This mattress can be fully inflated by adding triangles from DG to the inside triangulation of its B_e until all its non-manifold vertices have become manifold in the modified B_e. We call this first step in transforming BC_0 as the *inflating* operation and



(a) Point set (b) $EMST$ (c) BC_0 (d) B_{min}

Figure 3: Comparison of spanning graphs with constraints on vertex degree c: a) Point set from [DW01]. b) $EMST$ ($c \geq 1$). c) BC_0 ($c \geq 2$) with the inside of its enclosing boundary shaded. d) B_{min} ($c = 2$) with interior shaded.



(a) Figure 6 (b) Figure 1 (c) Figure 16

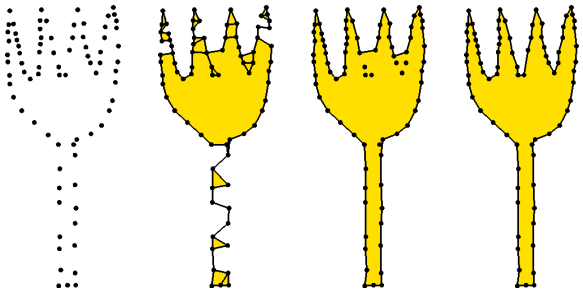
Figure 5: Sculpturing directly from the convex hull, as proposed by Boissonnat [Boi84], often fails, as can be seen from the results for some example point sets from this paper.

the result is denoted as inflated boundary complex BC_{infl}. Vertices in BC_{infl} therefore have degree of either 0 (*interior*) or 2 (on the B_e). Since the addition of triangles is also guided by the same minimization objective, and B_e is closed and manifold, BC_{infl} provides a better approximation to B_{min} than BC_0 (see Figure 6c)).

For the second step, we use the dual of the *inflating* operation, called *sculpturing*. Sculpturing removes triangles from the triangulation inside B_e until all interior vertices get exposed on the boundary, i.e., become part of B_e.

In Table 1 we compare the properties of the graphs described so far (see also Figure 3).

Boissonnat [Boi84] first introduced the term *sculpturing* as removing triangles with two vertices in the boundary (starting with the convex hull) in order to expose all interior vertices on that boundary. A criterion such as triangle circumradius was used to determine the order of removal. His algorithm is guaranteed to expose all vertices if a combinatorial search is used. However, with the convex hull as start set and using only heuristic sorting, it ends up quickly in local minima (see Figure 5). This is because there are points interior to the convex hull which are not reachable using just this heuristic.



(a) Point set (b) BC_0 (c) BC_{infl} (d) B_{min}

Figure 6: Area inside B_e always shaded: a) Point set. b) BC_0. c) After *inflating*: BC_{infl} with manifold boundary. d) After *sculpturing*: Interpolating boundary B_{out} (same as the output of TSP, as shown in [AM00]).

Our contribution to sculpturing is to appropriately choose the sorting criterion for shape characteristic minimisation, and to apply it starting from an already very closely approximated shape, namely the BC_{infl}. Experiments have shown that with this heuristic we do not get stuck into local minimum as frequently compared to starting with the convex hull. This can be seen from a number of examples presented in this paper (see Figure 5). An intuitive explanation for this is that B_e already includes some of those points that are interior to the convex hull.

Graph	Constraint
$EMST$	$c \geq 1$
BC_{min}	$c \geq 2$
BC_{infl}	$c = 0$, or $c = 2$
B_{min}	$c = 2$

Table 1: A comparison of the described graphs by their constraints of vertex degree c.

Given P, the entire algorithm consists of the following steps (see Figure 6):

- Compute DG of P.

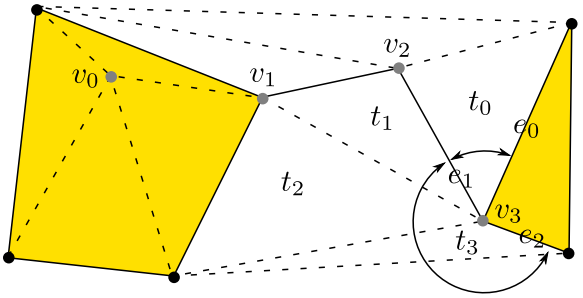


Figure 7: Vertex classification. Dotted edges are in DG, solid edges in BC_0, here equal to its B_e. Black dots are *manifold* (gray dots: v_0 is *interior*, v_1, v_2, v_3 are *non-conforming*). v_3 has two umbrellas (delimited by the arcs): e_0-e_1 bounds t_0, e_1-e_2 bounds the triangle fan $\{t_1,t_2,t_3\}$.

- Construct BC_0 from DG.
- Apply inflating operation to transform BC_0 into BC_{infl} consisting of a manifold boundary and interior vertices.
- Apply sculpturing operation to BC_{infl} to expose interior vertices onto the boundary to obtain B_{out}.

For the exact minimum boundary, the number of combinations of triangles which have to be considered in the process of inflating and sculpturing can be very large, and this is what makes our problem NP-hard. In our method, we use the heuristic of sorting triangle candidates based on the change in boundary length, $\Delta\|B\|$, in line with our objective of minimising length. This process results in an acceptable solution for the interpolating shape problem for a much larger class of point sets compared to previous methods. This class is defined later in section 6.2.

5.2. Inflating

We say an edge in BC_0 is reachable from the convex hull of the point set if:

1. it is an edge of the convex hull, or
2. it is an edge of a triangle in DG which has at least one edge on the convex hull, or
3. starting from a triangle in DG with an edge on the convex hull, it can be reached by traversing connected triangle edges, without traversing any other edge in BC_0.

All reachable edges of BC_0 make up B_e (see Algorithm 2).

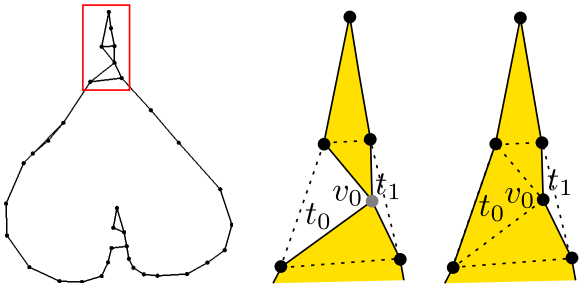
Let $T(B_e)$ denote all the triangles in $DG(P)$ which are inside B_e.

Lemma 2 The enclosing boundary B_e for BC_0 in DG contains all vertices in P either in B_e or in its interior.

Proof We start with B_e as the convex hull of P. So, B_e includes all $p_i \in P$ either on its boundary or in its interior. Removing a triangle from $T(B_e)$ never puts an edge $e_i \in BC_0$ into the outside of B_e. Since BC_0 spans all vertices in P, it follows that no $p_i \in P$ are outside of B_e. □

Input: BC_0
Output: B_e
Initialise B_e to convex hull of P;
while $e_i \in B_e \notin BC_0$ **do**
  t_i is triangle incident to e_i in $T(B_e)$;
  Remove t_i from $T(B_e)$
end

Algorithm 2: Determine enclosing boundary B_e for BC_0



(a) BC_0 (b) Detail (c) Inflated

Figure 8: a) BC_0 with single *non-conforming* vertex v_0 in the detail inside the frame. DG is shown as dotted and B_e as solid (interior shaded): b) t_0 and t_1 are both candidates at v_0. Since $\Delta\|t_0\|$ is the minimum, t_0 is selected to add to $T(B_e)$. c) v_0 is now manifold in B, therefore t_1 is no longer a candidate.

Lemma 3 The enclosing boundary B_e for BC_0 in DG is constructed in $O(n\log n)$ time using algorithm 2.

Proof The loop is executed at most for the $O(n)$ triangles in DG, testing its condition is $O(\log n)$ complexity as a set operation and therefore the complexity of the algorithm is $O(n\log n)$. □

We define an *umbrella* for a vertex v_i in B_e as its two incident edges as traversed in the given orientation. Note that a vertex may have more than one umbrella and in that case, those can share edges (see Figure 7).

Any vertex $v_i \in P$ can be classified by the number of umbrellas u which are formed by its incident edges in B_e (see Figure 7) as follows: v_i is *interior* to B_e if $u = 0$, *manifold* on B_e if $u = 1$ and *non-conforming* otherwise.

We define an *inflating-candidate* triangle for a B_e as a triangle t_i on its outside which is incident to a non-conforming vertex v_i in B_e.

The operation "Add a triangle t_i to $T(B_e)$" combines their space inside the new enclosing boundary B'_e which is formed by XORing the edges of t_i in B_e. $\Delta\|t_i\|$ provides a measure of the change in length to B_e. It is the value calculated by adding $\|e_j\|$ for all its edges $e_j \notin B_e$ and subtracting it for its edges $e_j \in B_e$ (see Figure 8). If the triangle is already in $T(B_e)$, then it leads to removal.

To *inflate* B_e we select and add a triangle from the current set of inflating-candidate triangles. This helps to reduce the number of its non-conforming vertices, since one is eliminated where a triangle is added to two edges in B_e with a shared non-conforming vertex. We repeat this triangle addition process until B_e becomes manifold. The selection criterion used is the smallest $\Delta\|t_i\|$, which helps to maximise the reduction in length of B_e. It gives priority to adding triangles which have already two edges in in B_e, as well as to those triangles which are largest and have the most acute angle at the non-conforming vertices.

Input: non-manifold B_e
Output: manifold B_e and BC_{infl}
Let $V(t)$ denote the set of three vertices of the triangle t and $V(DG)$ denote all vertices in DG;
Put all $v_i \in V(DG)$ which are *non-conforming* in V_{nc};
Every vertex v_i in V_{nc} has candidate triangles. This is so because of the following: if all its incident triangles were in B_e, it would have to be either interior or on the convex hull. Insert all candidate triangles t_i into a priority-queue PQ, sorted by ascending $\Delta\|t_i\|$;
while $PQ \neq \{\}$ **do**
  Remove first triangle t_i from PQ;
  Add t_i to $T(B_e)$ and update B_e accordingly;
  for $v_j \in V(t_i)$ **do**
    Update *conforming* state for v_j
  end
  for $v_j \in V(t_i)$ **do**
    for t_j *incident to* v_j **do**
      Determine candidate state of t_j and $\Delta\|t_j\|$;
      Remove, update or insert t_j in PQ
    end
  end
end
$BC_{infl} = B_e$ + (vertices in $V(DG)$ and not in B_e).

Algorithm 3: Inflating the enclosing boundary to a manifold

Lemma 4 Applying the *Inflating* operation as defined in Algorithm 3 to $B_e \in DG$ will always result in a manifold B_e.

Proof We first prove that the while loop is guaranteed to terminate. Any triangle $t_i \in DG$ which is on the outside of B_e is a candidate for addition to $T(B_e)$, if it is incident to a non-conforming vertex in B_e. All triangles outside B_e can at most be added once, since there is no removal operation. The while loop terminates if all vertices in B_e are manifold or in the limit all candidate triangles are added. In the latter case, B_e becomes identical to the convex hull of P, which is a manifold boundary. □

Lemma 5 Given an enclosing boundary B_e, the *Inflating* operation constructs BC_{infl} in $O(n \log n)$ time.

Proof Determining the *conforming* state of a vertex, if a triangle t_i is a candidate, and calculating its $\Delta\|t_i\|$ are all $O(1)$ complexity, since the computation is only dependent on the k incident edges in DG. The $O(n)$ triangles in DG are at most inserted a constant $3k$ times in PQ. The inner-most loop contains a constant number of $3k$ operations on sorted lists or sets. The algorithm is therefore of complexity $O(n \log n)$. □

5.3. Sculpturing

B_e of BC_{infl} is manifold, but BC_{infl} may contain some points of P as vertices interior to its B_e. As mentioned earlier, in [Boi84] Boissonnat defines that triangles with two vertices in the boundary may be removed. However, in order to keep B_e manifold, we have to apply the following conditions (derived from his 3D algorithm). Exactly one edge of the triangle must be in B_e and the vertex opposing that edge in the triangle must be interior. This guarantees that B_e remains manifold and all interior vertices can be exposed onto the boundary to transform B_e into any polygon in DT, which is also contained in B_e. It is easy to see that this holds even when we replace the convex hull with any manifold enclosing boundary, such as B_e in DG. Removal of the triangle exposes the interior vertex onto B_e since its incident edges become part of it, and this permits us to obtain any contained polygon in DG

Of course, the DG must contain a Hamiltonian cycle. However we have not encountered any point sets with non-Hamiltonian DG. Those have been observed to be extremely rare [Gen90] (see [Dil87] for a contrived example), it is therefore not a real concern in practice.

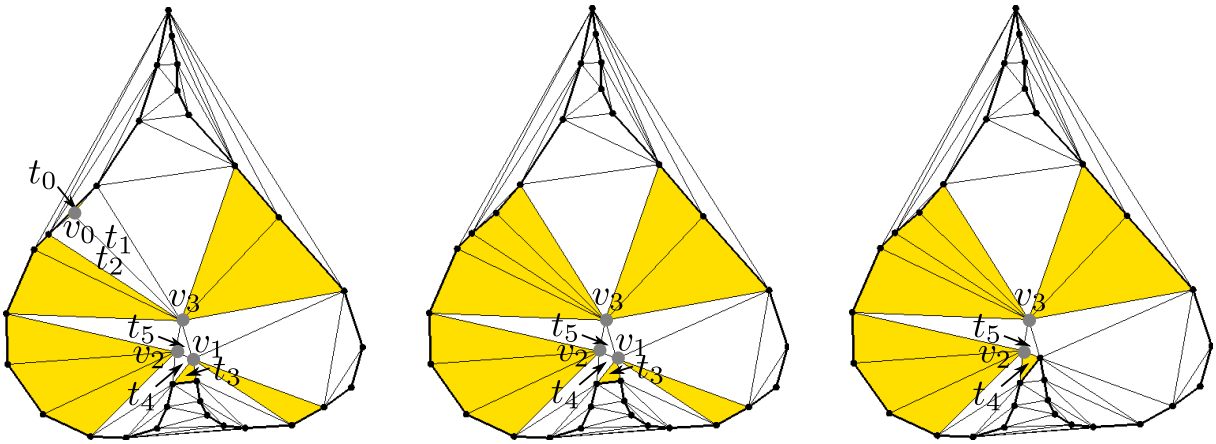
We define a *sculpturing-candidate* triangle for a B_e as a triangle t_i on its inside with one edge in B_e and its opposite vertex as interior in B_e.

Algorithm 4 exposes interior vertices efficiently to make them part of the interpolating boundary (see Figure 9).

Lemma 6 Sculpturing triangles from a boundary B_e using Algorithm 4 is $O(n \log n)$ complexity.

Proof Determining if a vertex is *interior* is $O(1)$ complexity and so are calculating $\Delta\|t_i\|$ for a triangle t_i and determining if it is a candidate. The outer loop is executed at most for the $O(n)$ triangles in DG. The first inner loop is run for the constant k incident edges in DG, the second one exactly twice. The algorithm is therefore of complexity $O(n \log n)$. □

To extract any contained polygon interpolating all points from a given B_e requires a combinatorial search. Since we limit our algorithm to $O(n \log n)$ complexity, sculpturing may not expose all points on the boundary for highly non-uniform point spacing. So the resulting boundary will still be manifold, but may not interpolate the entire point set. We have noted that such a case is limited to the local neighbourhood of these highly non-uniformly spaced points. Let C_u denote the class of point sets which do not contain such



(a) Initial (b) Exposed v_0 (c) Exposed v_3

Figure 9: Point set with manifold enclosing boundary B_e shown using thick lines, other edges in DG shown with thin lines, interior vertices shaded gray and triangle candidates for sculpturing shown shaded yellow. a) Of the 10 candidates, $\Delta\|t_0\|$ is minimal (t_0 is small and very thin). b) By removing t_0 from $T(B_e)$, interior v_0 becomes interpolated by B_e. At v_0, two new candidates have been added (t_1, t_2) as they now share edges with B_e. t_3 is selected next to remove, exposing v_1. c) t_4 and t_5 will be removed subsequently to interpolate v_2 and v_3 leading to B_{out}.

Input: BC_{infl}
Output: B_{out}
$B_{out} = B_e$ of BC_{infl};
Put all interior $v_i \in BC_{infl}$ in V_{int};
Insert all sculpturing-candidate triangles t_i into priority-queue PQ, sorted by ascending $\Delta\|t_i\|$;
while $PQ \neq \{\}$ **do**
  Remove first triangle t_i from PQ;
  $v_i \in t_i \notin B_{out}$;
  Remove t_i from $T(B_{out})$ and update B_{out} accordingly;
  for t_j *incident to* v_i **do**
    Remove t_j from PQ
  end
  for $e_j \in t_i$ *incident to* v_i **do**
    t_j is triangle in $T(B_{out})$ containing e_j;
    Determine candidate state of t_j and $\Delta\|t_j\|$;
    Insert t_j in PQ
  end
end

Algorithm 4: Sculpture to an interpolating boundary

highly non-uniformly spaced points. We shall discuss the relationship of this class with the sampling condition in the next section.

5.4. Performance

Theorem 1 The main algorithm in section 5.1 outputs a closed and non-intersecting manifold boundary in expected $O(n \log n)$ time for a point set P, provided DG contains a Hamiltonian cycle.

Proof The Delaunay triangulation step is of $O(n \log n)$ expected complexity as shown in [GS85]. All the following steps are also of $O(n \log n)$ complexity as proved in lemmas 1, 3, 5 and 6. The total complexity is therefore linearithmic.
Lemma 2 proves that all vertices in P are contained on or inside of B_e. By lemma 4 it produces a manifold boundary with interior points. Sculpturing rules maintain that manifold property and include all interior points on the boundary. □

We have observed that in practice the performance is determined to a large factor by the time for DG construction.

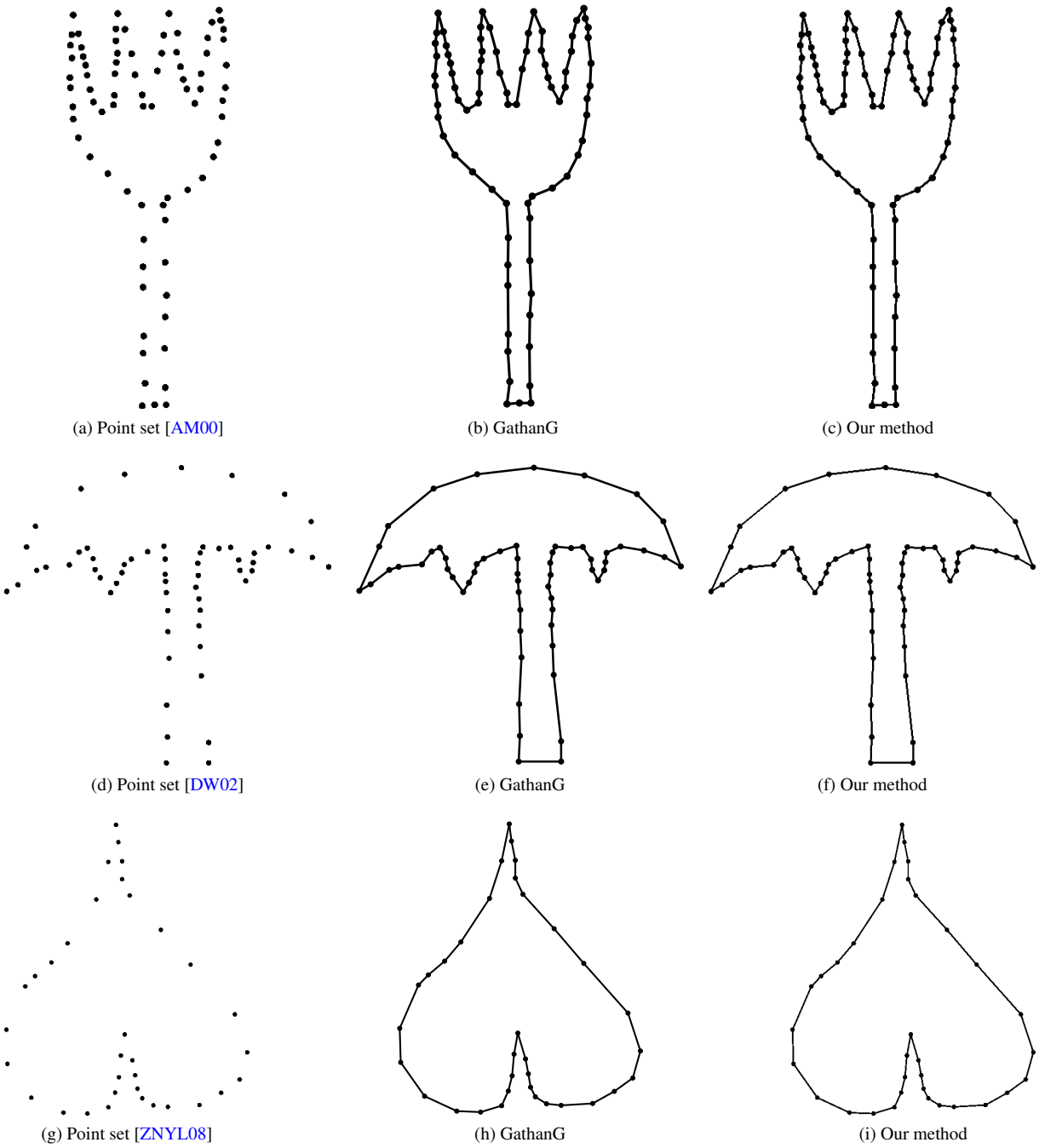
We note that for point sets in the class C_u, the resulting boundary also interpolates all points in P, since we defined it to guarantee interpolation in the last step, namely, sculpturing.

6. Results

The complete source-code for our technique has been made available online [Ohr11]. We use existing implementations for Delaunay triangulation [HS11] and disjoint sets [Ste11].

6.1. Comparison

We have tested our algorithm with a very large number of point data sets (some examples are shown in Figures 10, 11, 12 and 13). Many other algorithms also work for

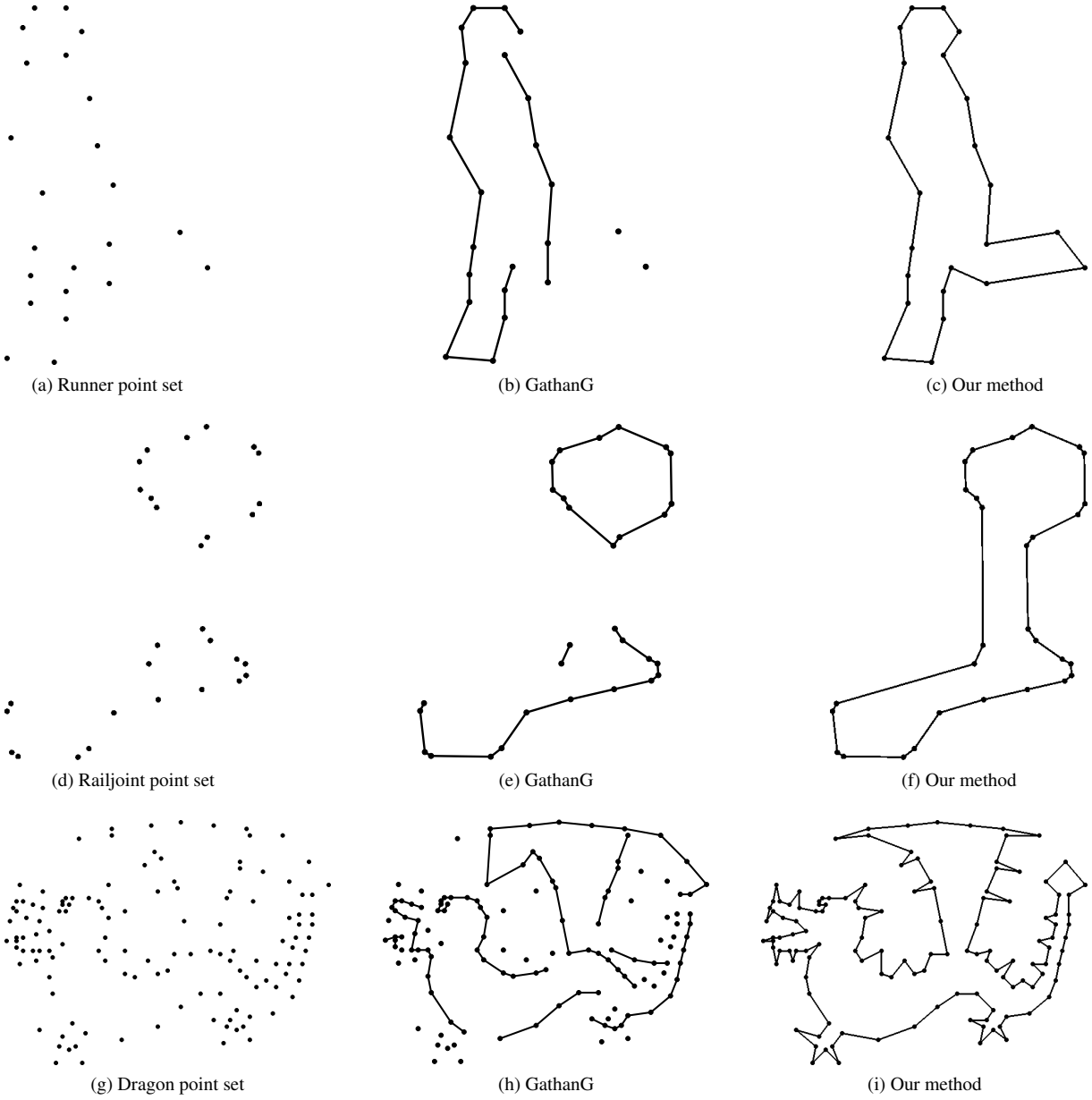


(a) Point set [AM00] (b) GathanG (c) Our method

(d) Point set [DW02] (e) GathanG (f) Our method

(g) Point set [ZNYL08] (h) GathanG (i) Our method

Figure 10: Boundary construction of nicely sampled point sets: Left column: Point set. Center column: *GathanG* with default parameters [DW02]. Right column: Our manifold result.

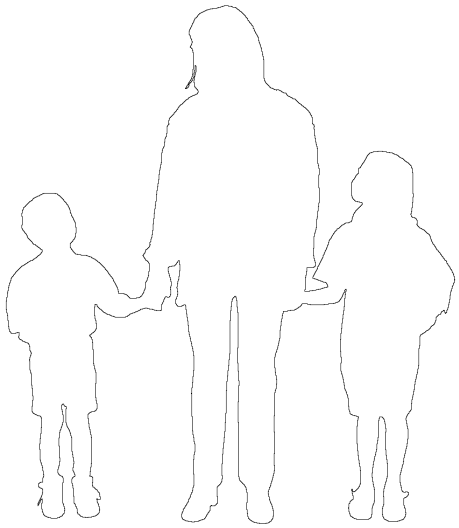


(a) Runner point set (b) GathanG (c) Our method

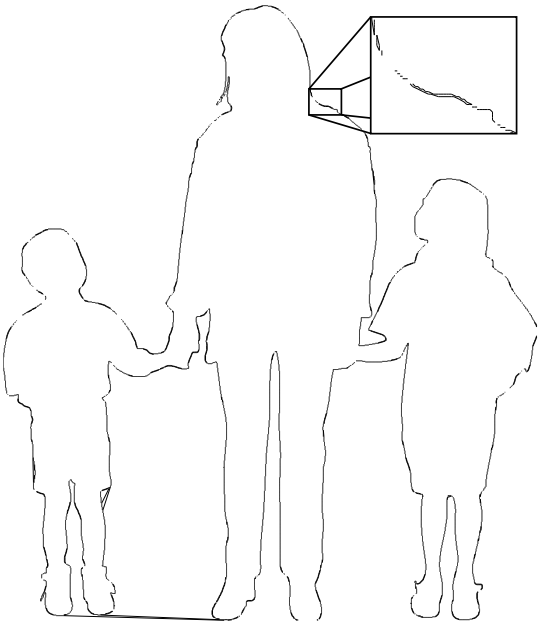
(d) Railjoint point set (e) GathanG (f) Our method

(g) Dragon point set (h) GathanG (i) Our method

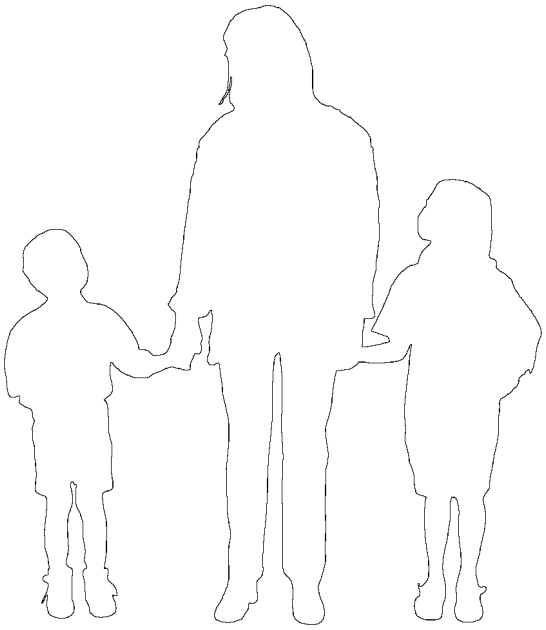
Figure 11: Boundary construction of challenging point sets: Left column: Point set. Center column: *GathanG* with default parameters [DW02]. Right column: Our manifold result. a-c) Point set sub-sampled from a silhouette video image. d-f) Railjoint, an engineering part [OM11]. g-i) Dragon point set, with many sharp corners.



(a) 10k points

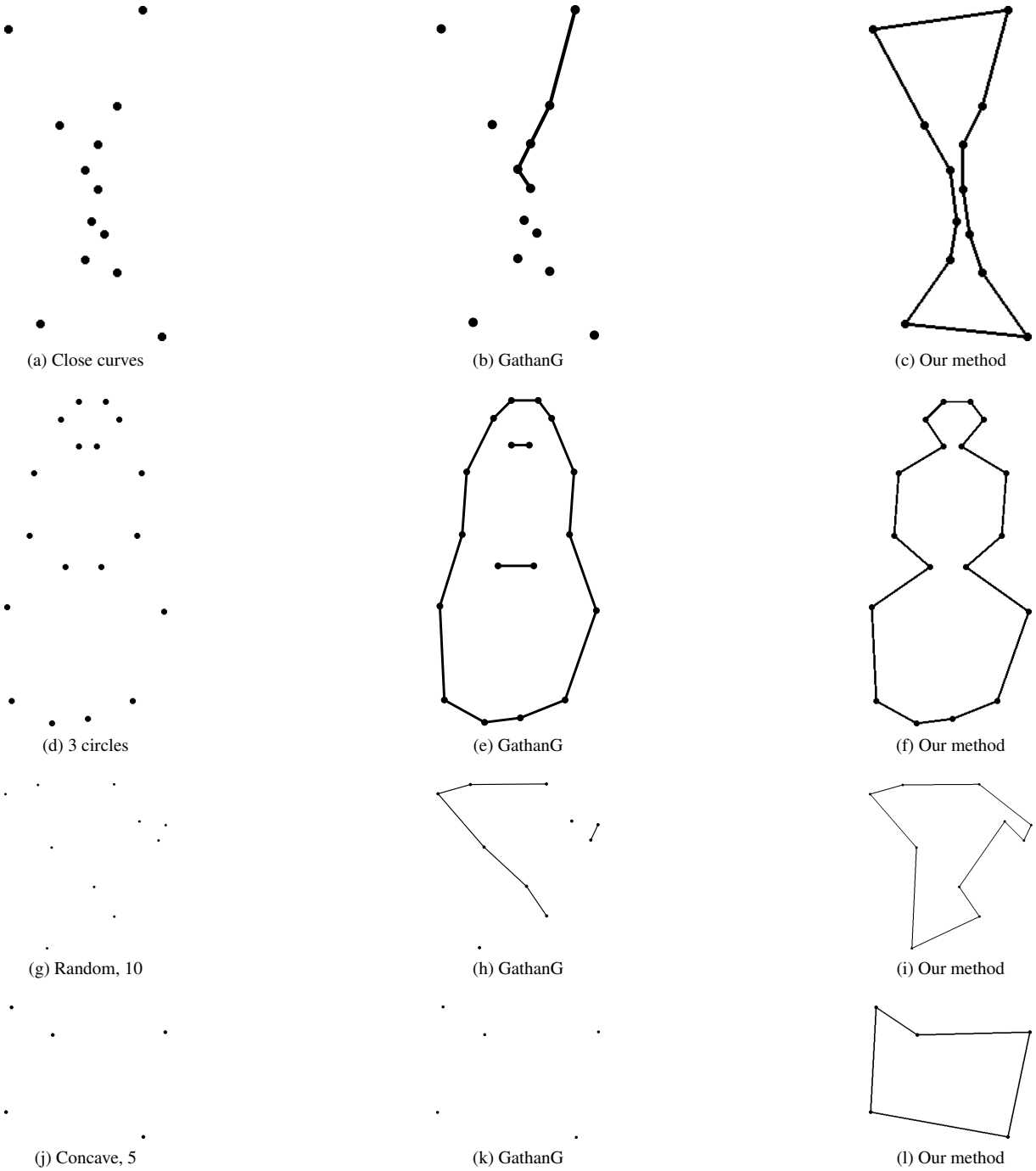


(b) GathanG



(c) Our method

Figure 12: 10000 points sampled from silhouette image. a) Point set. b) *GathanG* with default parameters [DW02]. Note the false connections, disconnections and doubled boundaries. c) Our method constructs a closed manifold, even for the extremely close boundaries.



(a) Close curves (b) GathanG (c) Our method

(d) 3 circles (e) GathanG (f) Our method

(g) Random, 10 (h) GathanG (i) Our method

(j) Concave, 5 (k) GathanG (l) Our method

Figure 13: Left column: Point set. Center column: *GathanG* with default parameters [DW02]. Right column: Our manifold result. a-c) Shape with extremely narrow portion. d-f) 10 random points. g-i) Three loops [OM11]. j-l) Concave polygon.

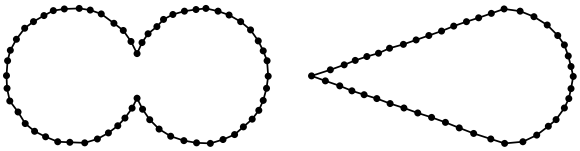


Figure 14: Our method reconstructs B_{out} (= B_{min}) for two point sets from [AMNS00], which all their TSP approximation algorithms fail to reconstruct.

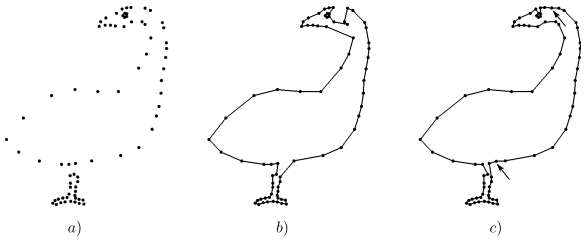


Figure 15: a) Goose point set from Amenta et al. [ABE98]. It does not represent a solid. b) The B_{out} constructed by our method is not B_{min}. c) After two Steiner points have been added to the point set (indicated by arrows), $B_{(}out) = B_{min}$ is constructed correctly.

typical large point sets such as Figure 12 which are usually uniformly and densely spaced. Hence, we focus on showing critical details of point sets which sampling-oriented reconstruction algorithms have not been able to handle correctly and efficiently (see Figure 13). We compare our results for a number of difficult point sets with *GathanG* from [DW02], which yields the best results of all earlier algorithms for such point sets (see Figure 10). [ZNYL08] has already compared their results with many of the other methods mentioned earlier. These results demonstrate that arbitrarily closely spaced boundary parts and very sharp corners as artifacts of sparsely and non-uniformly spaced points are all handled very well by our algorithm.

Figure 14 shows two point sets from [AMNS00] for which all of the six TSP-approximations algorithms listed in that paper fail. However, our method correctly constructs B_{min} for these point sets as well.

We discuss two point sets for which B_{out} is not the desired output. In the first case (see Figure 15) this is due to rather non-uniform point spacing in certain places. It is easy to see this, because after insertion of just two Steiner points, in the places where points are poorly spaced, our algorithm produces the desired result. In the second case (see Figure 16) our algorithm gets stuck in a local minimum. In both examples, the global minimum can be constructed by using an algorithm which does some combinatorial searching, such as in [OM11]. Since that algorithm tries to recover the exact B_{min} using a sequence of edge-swap operations, it exhibits

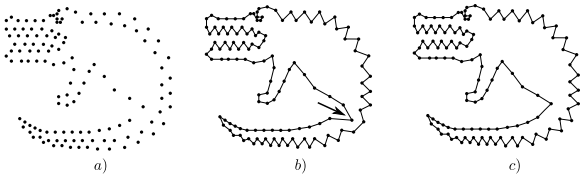
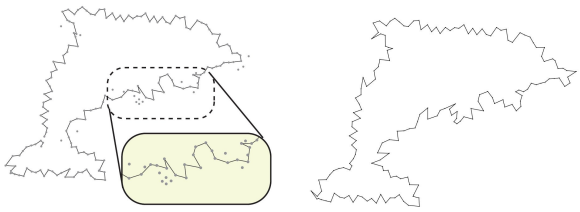


Figure 16: a) Crocodile point set from [OM11]. b) B_{out} construction by our method (has got stuck in a local minimum, see arrow). c) Exhaustive non-linearithmic search [OM11] yields B_{min}.



(a) Mehra et al. (b) Our method

Figure 17: Constructing the shape boundary from a noisy set of points. a) Figure presented in Mehra et al. [MTSM10]: It eliminates outliers rather arbitrarily, and does not fulfill its aim of producing a closed shape. b) Result of our method: B_{out} interpolates all the points as would be expected by humans.

linearithmic time only if there is a single such sequence. Under this strict constraint, it does work better for some highly non-uniform point sets. Yet, when this constraint is not met, it fails for many cases which this method reconstructs well (see for example Figure 10d). Further, the greater concern is that algorithm [OM11] can deteriorate quickly into combinatorial complexity.

6.1.1. Noisy Point Sets

Note that in this paper we only consider the problem of interpolating all the given points. For noisy point sets, any approximating technique which fits a curve close to the points may be considered a more appropriate choice. Fitting techniques however make implicit assumptions on an underlying curve, which especially for sparse sampling is not justified. Further these methods tend to perform low-pass filtering and may thus lose some essential information.

Noisy point sets represent measures taken of a shape at varying distance to its boundary. Features smaller than the noise threshold are invariably lost. On the other hand, if we consider the noisy point set as is, our method reconstructs the topology more robustly as compared to related work (see Figure 17).

De-noising can then be done based on a noise model as a post-processing step. This noise model can be constructed

on known measurement characteristics, or could simply be a probability distance function normal to an estimated boundary as a Gaussian distribution. The de-noised boundary can then be produced by fitting that model with the known connectivity of the reconstructed boundary with the original noisy data.

6.2. Sampling Condition

In order to construct a sampling condition for our algorithm, we first need to give some definitions:

The *medial axis* M for a smooth curve C is the closure of all points in $\mathbb{R}^2$ with ≥ 2 closest points in C [Blu67]. The *local feature size* $\mathrm{LFS}(x)$ for a point $x \in C$ is the Euclidean distance of x to its closest point $m \in M$ [Rup93]. $\mathrm{LFS}(x)$ is 1-Lipschitz continuous, which allows us to bound $\mathrm{LFS}(x)$ in terms of a sample p close to x, since $|\mathrm{LFS}(x) - \mathrm{LFS}(p)| \leq ||x,p||$.

Definition 2 refers to the widely used sampling condition given in [ABE98]:

Definition 2 A smooth curve C is said to be ε-sampled by a point set P if for each point $x \in C$ there exists a $p \in P$ such that $||x,p|| < \varepsilon\mathrm{LFS}(x)$.

In the discussion which follows let the ordered set $Q = \{q_0, q_1, q_2, ...\}$ denote the correct polygonal reconstruction of C using P, where $q_i \in P$.

Observation 1 Our experiments showed that the starting graph in our method, BC_0, already reconstructs a closed smooth ε-sampled curve C with $\varepsilon < 1$ for a large class of point sets P (see Figures 18 and 19 for some examples), even if P is locally non-uniformly sampled on C. Where B_{out} reconstructs C correctly, but BC_0 fails, is in the following situations:

- when parts of C with high curvature are close to each other and at the same time ε approaches 1, or
- C is sampled too locally non-uniform (as in all of Figures 10 and 13), such that points which are not adjacent on C are closer than their adjacent points.

Our method is able to reconstruct much more sparse point sets than others. It also works for a less restrictive ε-sampling with some constraint on local non-uniformity.

In order to be able to discuss these situations in a more formal way, we define a measure for local non-uniformity u as $e_{longer}/e_{shorter}$ for two adjacent edges in Q. With this we can define a sampling condition for which BC_0 itself results in the desired output B_{min}. Point sets fulfilling this sampling condition are therefore contained in the above-mentioned class C_u. It is important to note that C_u contains many more point sets outside of the above sampling condition for which applying the operations of inflating and sculpturing to BC_0 results in correct reconstruction.

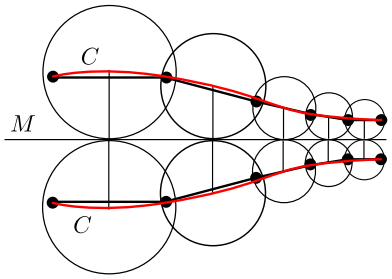
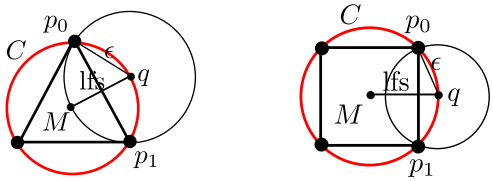


Figure 18: Two segments of BC_0 (thick black line) reconstructing the ε-sampled smooth curve C (shaded red) with ε just below 1, separated by its medial axis M. The circles represent local feature size as the distance between medial axis and the intersections at the half-sector of the curve segments between samples (length of the vertical lines).



(a) 3 samples (b) 4 samples

Figure 19: Red shaded smooth curve C (a circle) sampled with different number of points. Thick lines are edges in BC_0, M is the medial axis (here a point). q is the point on the segment p_0, p_1 of C with the largest distance from BC_0. $\|P_0, q\|$ divided by the local feature size LFS (=$\|q, M\|$) denotes ε for the ε-sampled C. a) 3 equi-distant point sample C with $\varepsilon = 1$. b) 4 equi-distant point sample C with $\varepsilon < 1$.

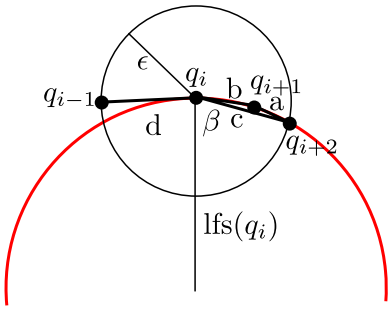


Figure 20: Illustration for deriving minimum conditions for samples $q_{i-1..i+2}$ to be arranged such that $d < c$, thus avoiding folding back in BC_0 as it happens in the example shown in Figure 21.

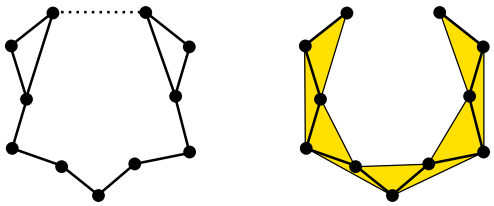
Theorem 2 BC_0 reconstructs a smooth ε-sampled curve C from the set of sample points P with $\varepsilon < 0.5$ and a local non-uniformity $u < 1.609$.

Proof We need to show that under the above sampling condition (i) $BC_0 = Q(= B_{out})$ and (ii) $BC_0 = B_{min}$.
(i) $BC_0 = Q$
If p is a point in P which is not adjacent to q_i in Q, then we shall call (q_i, p) a chord of Q.
Let us recall that the greedy construction of BC_0 takes up edges in order of length. If for every q_i, all chords (q_i, p) are longer than the two edges incident at q_i, then we are guaranteed that BC_0 equals Q. This is because the algorithm for constructing BC_0 will encounter the shortest two incident edges at every point before any chord and the algorithm terminates when every point has at least two incident edges. We prove next that under the given sampling condition, (q_i, p) is always longer than the two incident edges at q_i, namely (q_{i-1}, q_i) and (q_i, q_{i+1}).
Case 1) (q_i, p) does not intersect M: First we will show that in this case, chord length increases monotonically as we go forward along Q from (q_i, q_{i+2}), (q_i, q_{i+3}), and so on. We shall prove this by contradiction. Let (q_i, q_{i+k}) for some k be shorter or equal than (q_i, q_{i+k-1}). Let x be the point in the 1-disk $D \subset C$ between q_i and q_{i+k} with maximum length chord (q_i, x). Then the disk tangent to C at x with radius $LFS(x)$ has its center, which is in M, in that chord (q_i, x). Hence, either the chord (q_i, q_{i+k}) intersects M, contradicting our assumption, or it is part of the monotonically increasing sequence of chords in the other direction from q_i along C. (In the special case of a circle, for example, in which M is a single point, the chord length increases monotonically until it becomes the diameter and intersects M, and if it crosses over M, then it is part of the monotonically increasing sequence in the other direction.) Similar arguments can be made for monotonic increase in chord length along the other direction from q_i along Q, namely, (q_i, q_{i-2}), (q_i, q_{i-3}), and so on.
What remains to be shown is that (q_i, q_{i+2}) and (q_i, q_{i-2}) are always longer than (q_{i-1}, q_i) and (q_i, q_{i+1}). Then by the above monotonicity property, all chords will be longer than the 2 incident edges. We will only prove it for (q_i, q_{i+2}) as the proof for (q_i, q_{i-2}) is similar.
Lemma 10 in [ABE98] states that for an ε-sampling, the minimum angle spanned by three adjacent samples q_i, q_{i+1}, q_{i+2} in C is $\alpha = \pi - 4 \arcsin \frac{\varepsilon}{2}$. For $\varepsilon < 0.5$ this translates to $\alpha \approx 122^\circ$. By triangle inequality, the chord q_i, q_{i+2} cannot be shorter than the edge (q_i, q_{i+1}).
Next we will prove that the chord (q_i, q_{i+2}) cannot be shorter than the edge (q_{i-1}, q_i) under the above sampling condition. Assume a 1-disk $D \subset C$ between q_i and a point x such that no chord $(q_i, p \in D)$ intersects M, but the chord (q_i, x) does. Let sample $q_{i-1} \notin D$ and $q_{i+1}, q_{i+2} \in D$, or otherwise we are done. Let $a = |q_{i+1}, q_{i+2}|$, $b = |q_i, q_{i+1}|$, $c = |q_i, q_{i+2}|$, $d = |q_i, q_{i-1}|$ (see Figure 20) and therefore we want to prove that $d < c$.
$\frac{c}{d}$ is smallest when $\frac{b}{d}$, $\frac{a}{d}$ and their shared angle α become minimal. α is smallest if $q_{i..i+2}$ are lying on the circle tangent to q_i with radius $LFS(q_i) = 1$ without loss of generality (Lemma 7 in [ABE98] constrains the samples from being inside that circle) and angle β between the chord (q_i, q_{i+2}) and the normal to C at q_i is minimal. That is the case for $c = \varepsilon$ (Lemma 9 in [ABE98]). Because of the non-uniformity threshold u between adjacent edges, $\frac{b}{d} \geq \frac{1}{u}$ and $\frac{a}{b} \geq \frac{1}{u}$ and it follows that $d \leq u^2 a$. Hence we would like $c \geq u^2 a$.
$q_{i..i+2}$ form a triangle with lengths a, b, c. q_{i+2} lies at the intersection of a circle of radius ε with the circle containing the vertices. Since we have expressed a, b, c in terms of u, we can substitute these inequalities in the circle intersection equation and get $u < 1.60921$ for $\varepsilon = 0.5$.
Case 2) (q_i, p) intersects M: Let x be the point on the closed 1-disk of C between q_i and q_{i+1} that has maximum distance to both. We assume without loss of generality that $\mathrm{LFS}(q_i) \leq \mathrm{LFS}(p)$. Since (q_i, p) crosses M, $||(q_i, p)|| \geq 2\mathrm{LFS}(q_i)$. For the Euclidean distance d between q_i and q_{i+1}, applying the triangle inequality results in $d \leq ||q_i, x|| + ||x, q_{i+1}||$. We plug in $\varepsilon = ||q_i, x||/\mathrm{LFS}(x)$ from the above definition for ε-sampling and get $d \leq 2\varepsilon\mathrm{LFS}(x)$. Since LFS is 1-Lipschitz continuous, $|\mathrm{LFS}(x) - \mathrm{LFS}(q_i)| \leq ||x - q_i||$. The distance from x to q_i is $\varepsilon\mathrm{LFS}(x)$, so it follows that $\mathrm{LFS}(x) \leq \frac{\mathrm{LFS}(q_i)}{1-\varepsilon}$. We express $\mathrm{LFS}(x)$ through d and get $d \leq \frac{2\varepsilon}{1-\varepsilon}\mathrm{LFS}(q_i)$. This yields $d \leq ||q_i, p||$ if $\frac{\varepsilon}{1-\varepsilon} < 1$, which in turn is fulfilled for $\varepsilon < 0.5$.
(ii) Proof for $BC_0 = B_{min}$
Any other ordering of the points in Q, to obtain a closed interpolating curve, would require exchange of chords with incident edges. Since under the stated sampling condition, all chords are longer than incident edges, this would result in an increase in length. Hence $BC_0 = B_{min}$ under the above sampling condition.
□

With the above proof, our method guarantees reconstruction for much sparser point sets ($\varepsilon < 0.5$) as compared to other methods such as [ABE98] ($\varepsilon < 0.252$) and [DK99] ($\varepsilon < 1/3$). Since a higher ε permits sharper angles and closer points across the medial axis, this guarantee requires an additional restriction on local non-uniformity.

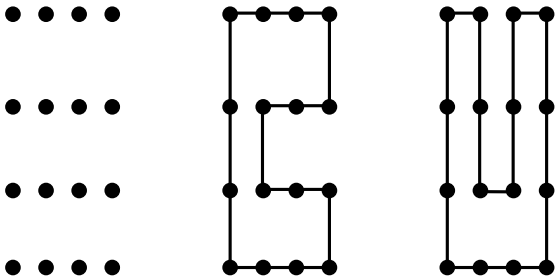
Let us recall that BC_0 is only our starting graph. The subsequent steps in our method which transform a non-manifold BC_0 into a manifold B_{out} enable handling of point sets beyond the above sampling condition. By enforcing the Gestalt law of closedness in obtaining B_{out}, our method is able to succeed with the reconstruction of much sparser sampling, like the cases in Figures 10 and 13 all of which go beyond the above sampling condition. This is because we believe that B_{min} reconstructs a smooth curve with far less stringent sampling requirements.

There are point sets which satisfy the ε-sampling condition but not the local non-uniformity requirement, and for which our transformation process may get trapped in a local minimum (see Figure 21 for a contrived example).



(a) BC_0 (b) B_{out}

Figure 21: a) BC_0 for a point set (noisy circle with hole at top) which samples a smooth curve with $\varepsilon < 0.5$ and in-between angle $> 122^\circ$, but violates the local non-uniformity condition $u < 1.75$. The dotted line is in B_{min}. b) Instead, inflating gets stuck in the local minimum B_{out}.



(a) Points (b) Short B_{out} (c) Long B_{out}

Figure 22: a) Point set sampling a smooth curve with $\varepsilon > 1$. b) B_{min}. c) Another possible B_{out}.

For $\varepsilon \geq 1$, Amenta et al. [ABE98] have observed (Observation 6) that there may not be a unique polygonal reconstruction of a smooth curve. Our method will choose one that is B_{min} (see Figure 22), following Gestalt laws.

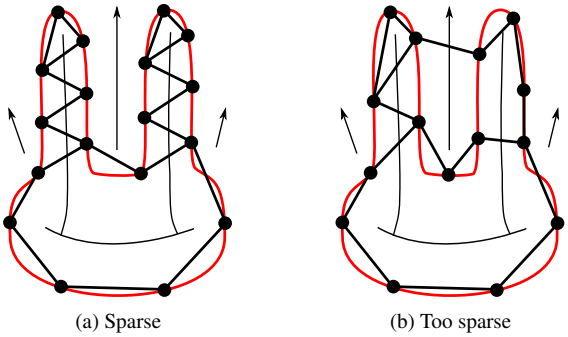
Conjecture 1 B_{min} reconstructs a smooth ε-sampled curve C for a set of points P with $\varepsilon < 1$.

Supporting arguments:

Theorem 12 in [ABE98] states that for such a curve C sampled by P, $DG(P)$ contains the edges which are adjacent on C. This would mean that our requirement that B_{min} be contained in DG is sufficient.

With $\varepsilon < 1$, from their Lemma 10 the minimum angle spanned by three adjacent samples on C is at least 60°.

Next, we have to show that Q is the minimum length closed interpolating curve with edges in $DG(P)$. Let P_{sub} denote the set of points belonging to a sub-sequence of edges in Q. Consider another non-intersecting sub-sequence of edges made up using all of P_{sub} to replace the original sub-sequence. Once again, we have two cases - the new sub-sequence either crosses the medial axis or it does not. If it does not, then under the condition that the angle between



(a) Sparse (b) Too sparse

Figure 23: The smooth curve C is shaded red. The thin curve is its medial axis (arrows imply continuing into infinity). Samples are connected by BC_0 (thick edges). a) C is sampled with $\varepsilon > 1$, but B_{out} still reconstructs C. b) C is sampled differently with $\varepsilon > 1$, but this time too sparsely for our method to reconstruct C with B_{out}. Note that there exist vertices whose incident edges intersect more than one connected component of the medial axis.

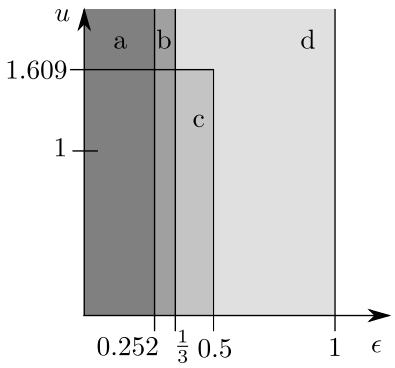


Figure 24: Comparing algorithms based on sampling criteria and guaranteed reconstruction: a) *Crust* [ABE98]. b) [DK99]. c) $BC_0 = B_{min}$. d) B_{min} (Our conjecture).

adjacent edges at every point in both the old and the new sub-sequence is at least 60°, it is clear that the replacement sub-sequence can only be longer than the original.

Since with $\varepsilon < 1$, the edge crossing the medial axis can be shorter than the original edge, we would require that the replacement subsequence is not shorter. We do not yet have the proof for this case. However, we have not been able to create a counter-example which shows that under this sampling condition, there exists a replacement sub-sequence which intersects the medial axis and is shorter in length. Hence this conjecture.

In Figure 24 we show how other sampling criteria guaranteeing reconstruction of smooth curves compare with ours. It illustrates that in order to reconstruct from less densely

spaced samples, some local uniformity has to be enforced. Note that we do not show the bound of *Gathan* [DW02] ($\varepsilon < 0.5$) for smooth parts of the curve, since it is not based on the same sampling criterion. It depends on the distance to both adjacent samples instead of just one and therefore the values for the sampling condition do not correlate to ours. In fact they prove in their Lemma 3 only a minimum angle of $\alpha = 150^\circ$ between incident edges to a sample, much larger than our $\alpha \approx 122^\circ$.

We also give an intuition for the limitations with which our method can reconstruct sparse sampling. In the examples we have worked with, we have noted that the inflating step can correctly reconstruct the curve locally where BC_0 is such that incident edges at a vertex intersect the medial axis only once (see Figure 23). In this situation, it can be seen well in Figure 13b that our method handles arbitrarily sparse sampling. How this relates to the sampling condition is not yet clear to us and will require more investigation.

7. Conclusion and Future Work

We propose a straight-forward and intuitive algorithm to extract the interpolating shape boundary present in an unorganised set of points. It handles more sparse point sets than earlier methods by formulating the boundary construction as a minimisation problem. A number of examples indicate that our method works better, particularly for sparse point sets with sharp corners in the shape. This has been shown by comparison with sampling-oriented methods, while guaranteeing competitive computational complexity. Experimental evidence shows that it may be sufficient to approximate the minimum total boundary length to yield an aesthetically pleasing curve (see Figure 16).

To guarantee a more relaxed point spacing criterion, in our future work, we will analyse the topology of BC_0 in order to segment it into components on which we can then determine where a different heuristic or a full search is required to find B_{min}.

Our work raises several interesting questions:

- Does there exist only a single B_{out} for a smooth $\varepsilon < 1$-sampled curve, and in that case, can the twice-differentiable 1-manifold be reconstructed faithfully?
- Can a sampling condition formulated using the two-dimensional parameter space of ε and u provide better guarantees with regard to sparsely sampled point sets?
- Is it possible to formalise the explicit condition under which sparse sampling is permitted, and under this condition also does there exist a single B_{out}?

More formal extensions to open and disconnected curves is also of interest.

Furthermore, based on some experimentation and promising initial results, we believe that our method will extend well to 3D point sets (see our overview presented in [OM12]).

Acknowledgements

This research was supported by Discovery Grants program of NSERC, Canada, Grand NCE and ENCS Faculty at Concordia. We thank our colleague Kaustubha Mendhurwar for providing us with silhouette point set data. We are grateful to the unknown reviewer who provided us with much simpler arguments for proving Theorem 2.

References

[ABCC11] APPLEGATE D., BIXBY R., CHVÃATAL V., COOK W.: *Concorde TSP Solver*, 2011 (accessed Mar 8, 2011). 3

[ABE98] AMENTA N., BERN M. W., EPPSTEIN D.: The crust and the beta-skeleton: Combinatorial curve reconstruction. *Graphical Models and Image Processing 60*, 2 (1998), 125–135. 2, 3, 13, 14, 15, 16

[AM00] ALTHAUS E., MEHLHORN K.: Tsp-based curve reconstruction in polynomial time. In *SODA '00: Proceedings of the eleventh annual ACM-SIAM symposium on Discrete algorithms* (Philadelphia, PA, USA, 2000), Society for Industrial and Applied Mathematics, pp. 686–695. 3, 5, 9

[AMNS00] ALTHAUS E., MEHLHORN K., NÄHER S., SCHIRRA S.: Experiments on curve reconstruction. In *Proc. 2nd Workshop Algorithm Eng. Exper* (2000), pp. 103–114. 3, 13

[Aro96] ARORA S.: Polynomial time approximation schemes for euclidean traveling salesman and other geometric problems. In *Journal of the ACM* (1996), pp. 2–11. 3

[Att97] ATTALI D.: r -regular shape reconstruction from unorganized points. In *Symp. Comp. Geom.* (1997), pp. 248–253. 2

[BB97] BERNARDINI F., BAJAJ C. L.: Sampling and reconstructing manifolds using alpha-shapes. In *In Proc. 9th Canad. Conf. Comput. Geom* (1997), p. 0. 2

[Blu67] BLUM H.: A Transformation for Extracting New Descriptors of Shape. In *Models for the Perception of Speech and Visual Form*, Wathen-Dunn W., (Ed.). MIT Press, Cambridge, 1967, pp. 362–380. 14

[Boi84] BOISSONNAT J.-D.: Geometric structures for three-dimensional shape representation. *ACM Trans. Graph. 3*, 4 (1984), 266–286. 5, 7

[Dil87] DILLENCOURT M. B.: A non-hamiltonian, nondegenerate delaunay triangulation. *Inf. Proc. Lett. 25*, 3 (1987), 149–151. 7

[DK99] DEY T. K., KUMAR P.: A simple provable algorithm for curve reconstruction. In *Proc. 10th ACM-SIAM SODA '99* (1999), pp. 893–894. 2, 15, 16

[DMR99] DEY T. K., MEHLHORN K., RAMOS E. A.: Curve reconstruction: Connecting dots with good reason. *In Proc. 15th ACM Symp. Comp. Geom 15* (1999), 229–244. 2

[DW01] DEY T. K., WENGER R.: Reconstructing curves with sharp corners. *Computational Geometry 19*, 2-3 (2001), 89 – 99. 2, 5

[DW02] DEY T. K., WENGER R.: Fast reconstruction of curves with sharp corners. *Int. J. Comp. Geom. Appl. 12*, 5 (2002), 353 – 400. 2, 9, 10, 11, 12, 13, 17

[EKS83] EDELSBRUNNER H., KIRKPATRICK D. G., SEIDEL R.: On the shape of a set of points in the plane. *IEEE Trans. Inf. Theor. IT-29*, 4 (1983), 551–559. 2

[FMG94] FIGUEIREDO L. H. D., MIRANDAS GOMES J. D.: Computational morphology of curves. *The Visual computer 11*, 2 (1994), 105–112. 2

[FS89] FREDMAN M., SAKS M.: The cell probe complexity of dynamic data structures. In *Proceedings of the twenty-first annual ACM symposium on Theory of computing* (New York, NY, USA, 1989), STOC '89, ACM, pp. 345–354. 4

[GB97] GLANVILL M., BROUGHAN K.: Curve and surface reconstruction in r2 and r3. In *HPC-ASIA '97: Proceedings of the High-Performance Computing on the Information Superhighway, HPC-Asia '97* (Washington, DC, USA, 1997), IEEE Computer Society, p. 395. 3

[Gen90] GENOUD T.: Etude du caractère hamiltonien de delaunays aléatoires. Travail de semestre, 1990. 7

[Gie99] GIESEN J.: Curve reconstruction in arbitrary dimension and the traveling salesman problem. In *Proc. 8th DCGI '99* (1999), pp. 164–176. 3

[GS85] GUIBAS L., STOLFI J.: Primitives for the manipulation of general subdivisions and the computation of voronoi. *ACM Trans. Graph. 4* (April 1985), 74–123. 8

[Hoo09] HOOS H. H.: *On the Empirical Scaling of Run-time for Finding Optimal Solutions to the Traveling Salesman Problem*. Technical Report 17, University of British Columbia, Department of Computer Science, 2009. 3

[HS11] HERT S., SEEL M.: dD convex hulls and Delaunay triangulations. In *CGAL User and Reference Manual*, 3.8 ed. CGAL Editorial Board, 2011. http://www.cgal.org/Manual/3.8/doc_html/cgal_manual/packages.html#Pkg:ConvexHullD. 8

[JT92] JAROMCZYK J., TOUSSAINT G.: Relative neighborhood graphs and their relatives. In *Proc. IEEE* (1992), pp. 1502–1517. 2

[KR85] KIRKPATRICK D. G., RADKE J. D.: A framework for computational morphology. *Comp. Geom.* (1985), 217–248. 2

[MTSM10] MEHRA R., TRIPATHI P., SHEFFER A., MITRA N. J.: Visibility of noisy point cloud data. *Computers & Graphics 34*, 3 (2010), 219 – 230. Shape Modelling International (SMI) Conference 2010. 13

[NZ08] NGUYEN T. A., ZENG Y.: Vicur: A human-vision-based algorithm for curve reconstruction. *Robotics and Computer-Integrated Manufacturing 24*, 6 (2008), 824 – 834. FAIM 2007, 17th Int. Conf. on Flexible Automation and Intellig. Manufact. 2

[Ohr11] OHRHALLINGER S.: Source-code for shape extraction algorithm, May 2011. https://sourceforge.net/p/connect2dlib/home/. 8

[OM11] OHRHALLINGER S., MUDUR S. P.: Interpolating an unorganized 2d point cloud with a single closed shape. *Computer-Aided Design 43*, 12 (2011), 1629–1638. 1, 3, 4, 10, 12, 13

[OM12] OHRHALLINGER S., MUDUR S.: Minimising Longest Edge for Closed Surface Construction from Unorganised 3D Point Sets. In *EG 2012 - Poster Proc.* (2012), pp. 25–26. 17

[Rup93] RUPPERT J.: A new and simple algorithm for quality 2-dimensional mesh generation. In *Proceedings of the fourth annual ACM-SIAM Symposium on Discrete algorithms* (Philadelphia, PA, USA, 1993), SODA '93, Soc. for Industr. and Appl. Math., pp. 83–92. 14

[Ste11] STEFANOV E.: *Disjoint Sets Data Structure Implementation*, 2011 (accessed May 13, 2011). http://www.emilstefanov.net/Projects/DisjointSets.aspx. 8

[Vel93] VELTKAMP R. C.: 3D computational morphology. *Computer Graphics Forum (EG '93) 12*, 3 (1993), 115–127. 2

[ZNYL08] ZENG Y., NGUYEN T. A., YAN B., LI S.: A distance-based parameter free algorithm for curve reconstruction. *Comput. Aided Des. 40*, 2 (2008), 210–222. 2, 9, 13