EUROGRAPHICS 2021
H. Rushmeier and K. Bühler
(Guest Editors)

Volume 40 (2021), Number 2
STAR – State of The Art Report

2D Points Curve Reconstruction Survey and Benchmark

S. Ohrhallinger[1] and J. Peethambaran[2] and A. D. Parakkat[3] and T. K. Dey[4] and R. Muthuganapathy[5]

[1]Institute of Visual Computing and Human-Centered Technology, TU Wien, Austria
[2]Department of Math and Computing Science, Saint Mary's University, Halifax, Canada
[3]Department of Computer Science and Engineering, Indian Institute of Technology Guwahati, India
[4]Department of Computer Science, Purdue University, Indiana, USA
[5]Department of Engineering Design, Indian Institute of Technology Madras, India

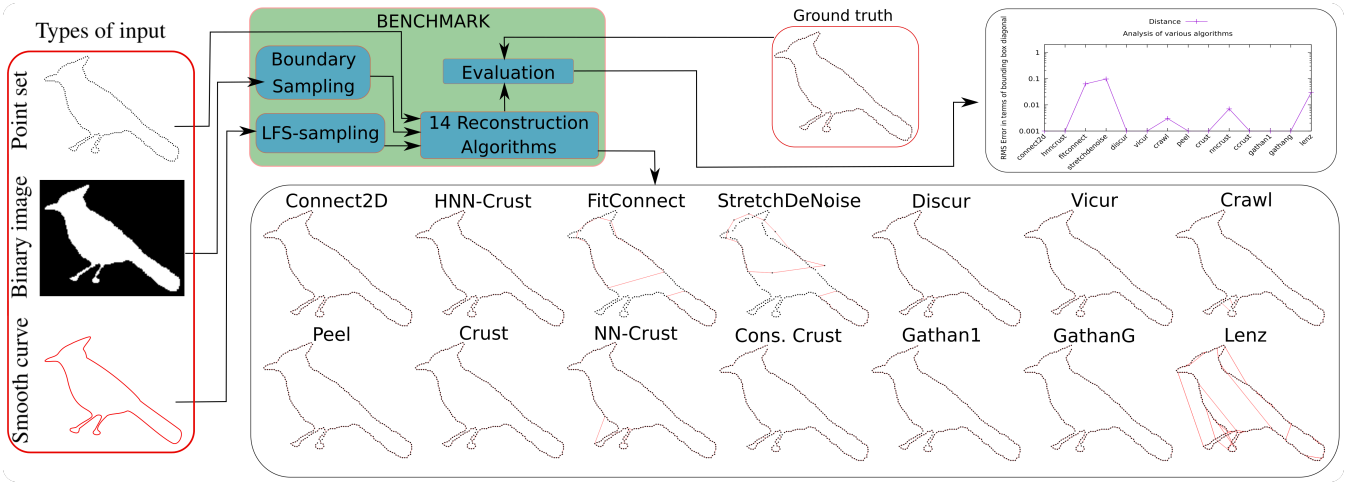

Figure 1: *We survey 36 curve reconstruction algorithms and compare 14 of these with quantitative and qualitative analysis. As inputs, we take unorganized points, samples on the boundary of binary images or smooth curves, and evaluate with ground truth.*

Abstract

Curve reconstruction from unstructured points in a plane is a fundamental problem with many applications that has generated research interest for decades. Involved aspects like handling open, sharp, multiple and non-manifold outlines, run-time and provability as well as potential extension to 3D for surface reconstruction have led to many different algorithms. We survey the literature on 2D curve reconstruction and then present an open-sourced benchmark for the experimental study. Our unprecedented evaluation of a selected set of planar curve reconstruction algorithms aims to give an overview of both quantitative analysis and qualitative aspects for helping users to select the right algorithm for specific problems in the field. Our benchmark framework is available online to permit reproducing the results and easy integration of new algorithms.

Categories and Subject Descriptors (according to ACM CCS): I.3.3 [Computer Graphics]: Picture/Image Generation—Line and curve generation

1. Introduction

Given a finite set of points P sampled from a planar curve Σ, recovering a polygonal approximation to Σ from P is generally known as curve reconstruction. Reconstruction of curves is a fundamental task in many applications such as reverse engineering of geometric models, outline reconstruction from feature points in medical imaging systems, and facial feature detection in the face recognition algorithms [ARZ05], among others, and an interesting problem by itself. Despite over three decades of tremendous research effort in the computational geometry, computer vision and graphics research communities, specific cases are still open in curve reconstruction, and there is no algorithm that would succeed on

all types of problems. Recent research trends, however, address specific aspects of reconstruction such as improved sampling conditions [OMW16], reconstructing from fewer number of samples and curves with sharp corners [Ohr13], reconstruction from unstructured and noisy point clouds [OW18a], a unified framework for reconstruction [MPM15], incremental labeling techniques for curve extraction [PPT*19], and applications of curve reconstruction to hand-drawn sketches [PM16].

A major hurdle to the ongoing efforts in designing new algorithms for curve reconstruction is the lack of a framework that provides a set of standard tools, algorithms and data for comparing and evaluating various reconstruction algorithms. Currently, algorithmic evaluations in this domain heavily rely on visual comparison. A meaningful practice in the empirical evaluation of reconstruction techniques is to compare the reconstructed results against the ground truth curves using error norms such as Hausdorff distance or $L2$-error norms. However, each research group has its own data set or generates the input data by sampling shapes from images. Such practices make it extremely difficult for researchers and practitioners from other fields to assess the performance of different curve reconstruction techniques and to conclusively determine a suitable algorithm for their scientific studies or applications. Furthermore, in most of the cases the algorithmic choices for the comparison are made based on the availability of implementations in the public domain.

To address these challenges, we have set up a benchmark framework for 2D curve reconstruction (https://gitlab.com/stefango74/curve-benchmark). It consists of a set of ***fifteen*** curve reconstruction techniques, including the recent ones, and a few support tools including a curve sampler that generates samples from smooth curves based on the ε-sampling [ABE98] criterion. Additionally, the benchmark provides a set of commonly used input data along with the ground truth curves for the experimental evaluation of algorithms in this domain. A set of newly generated input data that exhibits diverse features and is suitable for empirical studies is also included in the benchmark. Finally, we provide curve reconstruction evaluation criteria and features.

Besides presenting the curve reconstruction benchmark to the reader, this paper also covers principles and practices used in the curve reconstruction domain. We review the theoretical background, algorithms and their evolution, supporting tools, and evaluation criteria for curve reconstruction. The advantages and limitations of different methods are discussed. Apart from setting up the benchmark and reviewing various algorithms, the main contribution of this paper is an experimental assessment of the current state-of-the-art in the field with respect to standard error metrics such as Hausdorff distance, root mean square error (RMSE) and normal deviation. In the end, we delineate a few directions for future research.

1.1. Contributions

The main focus of this work is to review the available curve reconstruction literature and evaluate a competitive subset of curve reconstruction algorithms which take un-oriented and unorganized points as input and generate polygonal approximations to their underlying curves. We make the following key contributions.

- **Algorithms Review** A comprehensive review of the curve reconstruction literature up to date.
- **Benchmark** A benchmark consisting of prominent curve reconstruction algorithms, supporting tools, existing data and new test data along with the ground truth.
- **Evaluation** A thorough performance evaluation study comprising the algorithms provided in the benchmark. The study helps in demonstrating how the benchmark can be utilized for selecting the right curve reconstruction algorithm for a specific problem.

1.2. Related Work and Scope

To our knowledge, two prior works exist in the literature, and they both consider reconstruction of surfaces together with curves. The first is a comprehensive book [Dey06] which describes the basic theory leading to the development of the ε sampling condition and relating it to algorithms from the CRUST family with varying density before continuing to surface reconstruction algorithms, including noise and Morse theory there. A later concise report [KR14] adds some faster local, visual- and optimization-based methods. An empirical evaluation of a few early curve reconstruction algorithms is presented in [AMS00].

In this survey, we look in detail at curve reconstruction and recent developments, e.g. in terms of theoretical guarantees. In order to compare the algorithms and highlight their respective strengths, we have designed a benchmark for a comprehensive quantitative evaluation. In Table 1 we compare the capabilities of 36 curve reconstruction algorithms (categorized according to type) w.r.t. to input point sets requirements and output piece-wise curves.

1.3. Reconstruction Taxonomy

Boundary vs. Area Samples: In general, there are two types of inputs to the polygonal reconstruction problems: *boundary samples* and *area samples*. Boundary samples consist of points sampled along a curve while area samples are sampled across an entire region, including its boundaries as shown in Figure 3. While curve reconstruction is a well-defined problem, the reconstruction from area samples is ill-posed in nature [Ede98, PM15a]. The primary reason is a lack of precise mathematical definition for what constitutes the optimal approximation for the geometric shape of a point set with the points sampled from its interior. Furthermore, the shape perception from area samples is highly subjective since it often depends on a specific application context or human cognitive factors. A few unified algorithms [MPM15, DKWG08, GDJ*11, TPM20, TPM21] that handle sampled boundaries, as well as areas, have also been proposed. Since there are numerous algorithms such as α-shapes [EKS83] that handle both the input types with reasonable accuracy, we have not included unified algorithms in our experimental study. In this work, we focus only on reconstruction from boundary samples.

Implicit Vs. Explicit: Broadly, the reconstruction techniques can be grouped into implicit fitting and explicit reconstruction. Implicit methods attempt to define a smooth function $f: R^2 \rightarrow R$ such that the zero level set of f approximates the underlying curve in the input points as illustrated in Figure 2. Explicit reconstruction deals with connecting the input points using edges or triangular faces

Capabilities:	Param	Input			Output					
Algorithm	count	n.-u.	noise	outl.	manifold	open	mult.	sharp	guar.	time complexity
Graph Based:										
α-Shapes [EKS83]	1	no	no	no	yes	no	yes	no	yes	O(n log n)
β-skeleton [KR85]	1	yes	no	no	no	no	yes	yes	no	O(n log n)
γ-neighborhood [Vel93]	2	yes	no	no	no	no	yes	yes	no	O(n log n)
EMST-based [FMG94]	0	no	no	no	yes	only	no	no	yes	O(n log n)
Ball-pivoting [BB97]	1	no	no	no	yes	no	yes	no	yes	O(n log n)
r-regular shapes [Att97]	1	no	no	no	no	no	yes	no	yes	O(n log n)
Edge exchanging [OM11]	0	yes	yes	no	yes	no	no	yes	no	NP
CONNECT2D [OM13]	0	yes	yes	no	yes	no	no	yes	yes	O(n log n)
Shape-hull graph [PM15b]	0	yes	no	no	yes	no	no	yes	no	O(n log n)
Voronoi Labeling [PPT*19]	0	yes	no	yes	yes	no	yes	yes	yes	O(n log n)
CRAWL [PM16]	0	yes	no	yes	no	yes	yes	no	no	O(n log n)
Feature Size Criteria:										
CRUST [ABE98, Gol99]	0	yes	no	no	yes	no	yes	no	yes	O(n log n)
NN-CRUST [DK99]	0	yes	no	no	yes	yes	yes	no	yes	O(n log n)
CONS. CRUST [DMR99]	0	yes	no	yes	no	yes	yes	no	yes	O(n log n)
[Len06]	2	yes	no	no	no	yes	no	yes	yes	O(n log n)
[Hiy09]	0	yes	no	no	yes	no	yes	no	yes	O(n^2 log n)
HNN-CRUST [OMW16]	0	yes	no	no	yes	yes	yes	no	yes	O(n log n)
Noisy Points Fitting:										
[Lee00a]	1	yes	yes	yes	yes	yes	no	no	no	O(n^2)
[CFG*05]	2	yes	yes	no	yes	yes	yes	no	yes	O(n log n)
ROBUST HPR [MTSM10]	5	yes	yes	no	yes	yes	yes	yes	no	O(n log n)
[Rup14]	1	yes	yes	yes	yes	yes	yes	no	no	O(Mn)
[WYZ*14]	4	yes	yes	yes	no	yes	yes	yes	no	O(d log d)
FITCONNECT [OW18a]	0	yes	yes	yes	yes	yes	yes	yes	yes	O(nk^2)
STRETCHDENOISE [OW18b]	0	yes	yes	yes	yes	yes	yes	yes	yes	O(nk^2)
Sharp Corners:										
[FR01]	8	yes	no	no	no	yes	yes	yes	yes	O(n log n)
GATHAN [DW01]	1	yes	no	no	no	yes	yes	yes	no	O(n log n)
GATHANG [DW02]	1	yes	no	no	no	yes	yes	yes	yes	O(n log n)
Traveling Salesman:										
[Gie99]	0	no	yes	no	yes	no	no	yes	yes	O(n log n)
[AM00]	0	yes	yes	no	yes	no	no	yes	yes	O(n log n)
[Aro98]	0	yes	yes	no	yes	no	no	yes	yes	O(n (log n)$^{O(c)}$)
Concorde solver [ABCC]	0	yes	yes	no	yes	no	no	yes	TSP	NP
Non-manifold:										
Opt. Transp. [DGCSAD11]	0	yes	yes	yes	no	yes	yes	no	yes	O(n log n)
PEEL [PMM18]	2	yes	yes	yes	no	yes	yes	no	yes	O(n^2)
EC-SHAPE [MKPM17]	0	yes	no	no	yes	no	no	no	yes	O(n log n)
HVS Based:										
DISCUR [ZNYL08]	0	yes	no	no	yes	yes	yes	yes	yes	O(n log n)
VICUR [NZ08]	4	yes	no	no	yes	yes	yes	yes	no	O(n log n)

Table 1: *Algorithms grouped by categories, with their input and output capabilities (n.-u. = non-uniform, guar. = sampling condition for manifold reconstruction)*

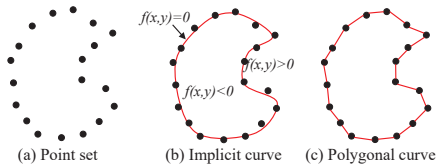

Figure 2: *Illustration of implicit function in 2D and an example of explicitly reconstructed polygonal curve.*

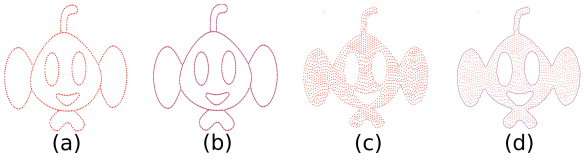

Figure 3: *Two common types of inputs to the 2D reconstruction algorithms. (a) Boundary sample (b) Reconstructed curves (c) Area sample (d) Reconstruction from area samples.*

subjected to certain geometric criteria, which results in a piece-wise linear approximation to the underlying curve. Implicit reconstruction algorithms utilize the orientation of the point sets to define the curve function. The orientation of the points is obtained through point normals or partially inferred through the segmentation of binary images (see Figure 1). On the contrary, explicit reconstruction algorithms take un-oriented point sets. A vast majority of explicit methods interpolate all the input points including noisy data or outliers. As a result, explicit curve reconstruction algorithms are not robust to noise unless additional algorithmic criteria such as the ones in [WYZ*14, OW18a, OW18b] to deal with the noise/outliers are included. Implicit methods work on noisy data. However, since the iso-curves are extracted using marching squares on quadtrees, an appropriate quadtree depth has to be determined and preset in the case of implicit methods. Highly detailed curves can be generated for larger depth values, however, at the expense of increased computational time. While popular in 3D surface reconstruction, we have not found implementations or results for curve reconstruction, and so this category is not included in our comparison or evaluation. However, we have dedicated a section (Section 4) to review the popular implicit fitting algorithms.

Categorizing the algorithms is a difficult task since some of them could fit with several aspects. We propose the following taxonomy to subdivide them by their characteristics we deemed most important:

- *Graph Based*: constructs a graph from the points and then filters the outline by some criterion
- *Feature Size Criteria*: differing approaches, but the required sampling density is proven in relation to local feature size
- *Noisy Points Fitting*: able to recover the original smooth curve from noisy samples
- *Sharp Corners*: can reconstruct angles $< 90^\circ$ degrees instead of smoothing them over
- *Traveling Salesman*: minimizes total curve length
- *Non-manifold*: also handles (self-)intersections in curves
- *HVS Based*: inspired by the Gestalt laws on how Human eyes perceive visual elements

This taxonomy is also used to structure the descriptions of algorithms in Section 3.

We examine the algorithms for the following properties:

Input point set:

- *Non-uniformity*: no uniform point spacing required
- *Noise*: samples can be displaced from the original curve
- *Outliers*: additional points far from the curve are ignored

Output piece-wise linear curve:

- *Manifoldness*: each vertex has ≤ 2 incident edges
- *Open curves*: end vertices (=holes in boundary) can exist
- *Multiply Connected*: curve has 2 or more components
- *Sharp corners*: angles $< 90^\circ$ can be reconstructed
- *Guarantees/Conditions*: for successful reconstruction
- *Time complexity*: worst-case behavior of the algorithm.

In the following sub-section we detail challenges of these properties:

1.4. Challenges for Reconstruction

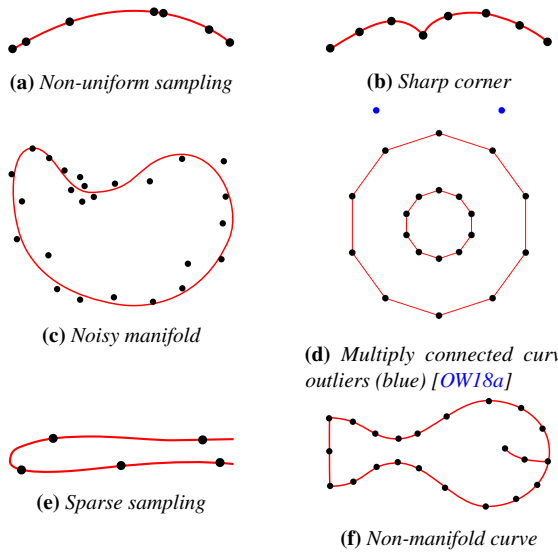

Figure 4: *Various challenging input point configurations for curve reconstruction.*

Here we give a list of challenging aspects in curve reconstruction (see Figure 4) - all of the input configurations can also be found combined:

Non-uniformity: Some algorithms require a globally uniform maximum spacing between samples. The disadvantage is that only features larger than that distance can be reconstructed. Algorithms which can reconstruct from non-uniform sampling (see Figure 4a) are then not limited to a specific absolute size for reconstructing features, but only restricted by a too sparse sampling of features (see Figure 4e). A limitation can also be given for relative uniformity as a factor between spacings of adjacent samples.

Noise: When sensing data, such as silhouettes of objects, samples are usually perturbed by noise from measurement errors (see Figure 4c). Algorithms that strictly interpolate the input points will, in the best case, reconstruct a locally perturbed curve. Instead, points can be fitted in order to recover the original curve either by smoothing over the noise, or by denoising with knowledge/estimate of the local noise extent.

Outliers: Sensing data can also introduce erroneous points far from the original curve. These are labeled outliers (see Figure 4d), should not be considered in the reconstruction, and must thus be classified and excluded beforehand.

Manifoldness: A boundary of an object is always a manifold curve, i.e., points are assigned at most two neighbors (see Figure 4c). Otherwise, the curve can become self-intersecting (see Figure 3a), which is useful for reconstructing drawings.

Open curves: Curves can be open if samples are missing from an object boundary, and still be manifold, also a collection of open curves, e.g., drawings (see Figure 4a) .

Multiply Connected: If boundaries of more than one object are to be reconstructed, these must not be interconnected in order to remain manifold. Holes in shapes are homotopy equivalent to that (see Figure 4d).

Sharp corners: Angles that are below 90 degrees are more difficult to reconstruct as it is more ambiguous which neighbors to connect if they are not in the opposite half-space of the other neighbor (see Figure 4b). Also, fitting algorithms tend to round off such sharp corners.

Guarantees/Conditions: It is very useful to know to which extent a curve can be reconstructed from a sampling. Guarantees can be given in terms of uniformly spaced sampling as a maximum global distance value or relative factor between neighbor points, a sampling condition in terms of the feature size (e.g., ε-sampling), percentage of outliers, the extent of noise range and statistical distribution, and as a distance of the reconstruction from the original curve.

Time complexity: Since points in a plane are mostly a small number, such as a few thousand, optimization is not critical, but worst time complexity matters, as some algorithms have $O(n^2)$ or are not solvable in polynomial time.

2. Preliminaries

Many of the following definitions were introduced in this seminal paper [ABE98]. We follow that up with an overview of proximity graphs which are used in many algorithms.

2.1. Definitions and Notations

Let P be a set of n points sampled from a simple closed planar curve Σ. The curve Σ (if closed) is said to be *convex* if the line segment between any two points on the curve falls in the interior, $I(\Sigma)$. Otherwise, it is *concave*. The curvature κ at a point p of Σ is the rate of change of direction of the tangent line at p with respect to arc length s. An inflection point (*IP*) on the curve is a point where $\kappa = 0$ but $\kappa' \neq 0$. Concave subsets of a curve are characterized by the sign of the local curvature κ. Concave subsets exist between two inflection points and have a negative local curvature sign ($\kappa < 0$) [PM15b].

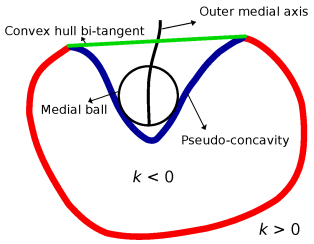

Figure 5: *Illustration of pseudo-concave subsets of a simple closed curve in 2D. The blue curve segment constitutes its pseudo-concave subset. Image courtesy [PPT*19].*

Let E be the set of all open, connected regions of $Convexhull(\Sigma) \setminus \Sigma$. Each region given by the closure E is defined as a pseudo-concave region (PCR) of Σ (Figure 5). The subset of Σ in each PCR is called a *pseudo-concavity*. The edges of the convex hull of Σ in each PCR are called convex hull bi-tangents. Based on the radii of medial balls, Peethambaran et al. [PM15b] define *divergent pseudo-concavity* for simple closed planar curves. A pseudo-concave subset of a curve Σ is *divergent* if the radii of medial balls monotonically increase as they go along the outer medial axis from one end to the convex hull bi-tangents' end. The curve Σ is said to be divergent if all its pseudo-concave subsets are divergent.

From Section 4 in this paper [OMW16] we repeat the following definitions:

The *medial axis* M of Σ is the closure of all points in $\mathbb{R}^2$ with two or more closest points in Σ [Blu67]. A *medial ball* $B(c,r)$, centered at $c \in M$ of Σ with radius r, is a maximal ball whose interior contains no points of Σ.

A smooth curve C (as opposed to Σ, which may contain inflection points and sharp corners) is a (collection of) bounded 1-manifold(s) embedded in $\mathbb{R}^2$, which are twice-differentiable everywhere except perhaps at boundaries [DT14].

We define the *local feature size* $\text{lfs}(p)$ for a point $p \in C$ as the Euclidean distance from p to its closest point m of M. This definition is loosely based on [Rup93], but simplified because we are only considering smooth curves.

Definition 1 is a widely used sampling condition [ABE98] that captures features regardless of size as opposed to globally uniform sampling distances:

Definition 1 A smooth curve C is ε-*sampled* by point set S if every point $p \in C$ is closer to a sample than an ε-fraction of its local feature size: $\forall p \in C, \exists s \in S : \|p,s\| < \varepsilon \,\text{lfs}(p)$.

In contrast, the *reach* [Fed59] for a set $\mathcal{S}$ is the largest "radius" r such that points closer than r to $\mathcal{S}$ have a unique closest point of $\mathcal{S}$. The reach is similar to the smallest distance to the medial axis. This inspires our definition of the *reach* of a curve interval I as $\inf \text{lfs}(p) : p \in I$, where the lfs is defined by all of C. [OMW16]

2.2. Proximity Graphs

In general, proximity graphs such as the relative neighborhood graph (RNG), Gabriel graph, Sphere-of-Influence graph [Tou88], and β-skeletons [KR85] play a vital role in defining the shape and structure of planar point sets, including curve samples [JT92]. Since many reconstruction algorithms and accompanying theory [EM94, Vel92, FMG94, Boi84a] are built around proximity graphs, it seems appropriate to formally define these proximity structures for a better review and understanding of various reconstruction algorithms.

The *relative neighborhood graph* consists of edges (p,q) such that $d(p,q) \leq d(p,x)$ and $d(p,q) \leq d(q,x)\ \forall\, x \in P$ where $x \neq p$ or q [JT92]. Varying the size (β) of the region of influence of each pair of points in RNG generates a set of neighborhood graphs called β-*skeletons* [KR85]. In β-skeletons, the neighborhood $U_{p,q}(\beta)$ of two points p and q for any fixed β ($0 \leq \beta \leq \infty$) is defined as the intersection of two spheres [JT92] as follows:
$U_{p,q}(\beta) = B((1-\frac{\beta}{2})p + \frac{\beta}{2}q, \frac{\beta}{2}d(p,q)) \cap B((1-\frac{\beta}{2})q + \frac{\beta}{2}p, \frac{\beta}{2}d(p,q))$.

A *minimal spanning tree* MST(P) of P is a tree (a cycle-free graph) that spans all the points in P with the least total cost of edge weights. The *Gabriel graph* of P contains an edge (p,q) if and only if the ball passing through p and q centered at the edge (p,q) is empty [JT92]. A triangulation of P is a subdivision of the plane by edges between vertices in P such that no edge connecting two vertices in P can be added in the plane without creating a self-intersection.

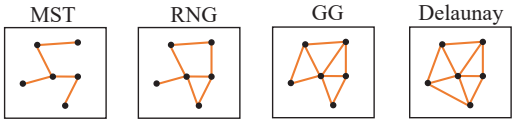

Figure 6: *Examples of proximity graphs for a point set.*

A Delaunay triangulation is a triangulation $D(P)$ such that no point in P is inside the circumcircle of any triangle in $D(P)$. A graph is a *Delaunay graph* if it is the Delaunay triangulation of some set of points in the plane. The Delaunay graph has many interesting properties that make it a central data structure in many curve reconstruction algorithms. The relationship among different proximity graphs, i.e., $MST(P) \subseteq RNG(P) \subseteq GG(P) \subseteq D(P)$ (see Figure 6) is a well-established result [JT92].

For $p \in P$, let r_p be the minimum distance from p to $P \setminus p$, and let $B(p,r_p)$ denotes the open ball of radius r_p centered at p. Then there exists an edge (p,q) in the *sphere of influence graph* of P if and only if $d(p,q) \leq r_p + r_q$ [Tou88]. These definitions may result in a graph rather than a simple polygon and may contain disconnected regions, non-manifold edges and vertices.

3. Explicit Reconstruction of Curves

In this section, we describe the algorithms, grouped by categories. The first set of reconstruction algorithms (see Subsection 3.1) were developed *based on the proximity graphs* mentioned above. Then, the introduction of a *feature size based sampling condition* permitted to reconstruct features of arbitrary size (see Subsection 3.2). These algorithms were then extended to handle *noisy samples* (see Subsection 3.3) and *sharp corners* (see Subsection 3.4). The *traveling salesman problem* (see Subsection 3.5) solves a special case of curve reconstruction. Some algorithms can even reconstruct *self-intersecting curves* (see Subsection 3.6), which allows for new applications. The Gestalt laws of perception led to *curve generation based on the Human Visual System* (see Subsection 3.7). Finally, curves can be fitted approximately to an *implicit function* (see Section 4) describing the underlying curve.

3.1. Graph-Based Reconstruction

Jarvis [Jar77] was the first to develop a notion of shape for a dense unorganized point set in a plane.

This was later formalized by Edelsbrunner et al. [EKS83]. They defined α-shapes as a generalization of the *convex hull*, which permits replacing edges shorter than a globally uniform constant with the opposite two edges of the containing triangle in the Delaunay triangulation of the points.

Edelsbrunner and Mücke [EM94] later extended this concept to $\mathbb{R}^3$. It can intuitively be understood as a mass of soft ice cream (the convex hull) containing hard chocolate chips (the points) from which a ball-shaped spoon with radius α nibbles off the ice cream where it can move freely between the chocolate chips, leaving the α-shape (which is equal to the convex hull for the case $alpha = \infty$).

Bernardini and Bajaj [BB97] used that definition to design a construction algorithm for α-shapes in $\mathbb{R}^2$: It rotates a disk of radius $\sqrt{(\alpha)}$ around a point until it touches another point which is then connected by an edge and continues with the new point until a loop is created.

Later Bernardini et al. [BMR*99] developed an extension to $\mathbb{R}^3$, the *ball-pivoting* algorithm.

For point sets that are sampled on a curve (as opposed to dense sampling inside the shape as well), edges reconstructing this curve can be selected using the definition of the β-skeleton [KR85]: All edges of the Delaunay triangulation of the point set that are shorter than $\frac{\beta}{2}$ times the radii of the circumcircles of their adjacent triangles belong to this β-skeleton.

Veltkamp proposed the γ-neighborhood graph [Vel92], which unifies the convex hull, the Delaunay triangulation, the *Gabriel graph*, and the β-skeleton, and presents the relations to and between other neighborhood graphs.

Based on the γ-neighborhood graph, he later showed that a greedy algorithm based on both local and global measures could also reconstruct boundary polygons that are not in the Delaunay triangulation [Vel93].

This was necessary because Delaunay triangulations do not always contain a *Hamiltonian cycle*, a simple polygon interpolating all its points, although that case has been shown to be very rare [Gen90].

Boissonat [Boi84a] used *sculpturing* to replace edges in the convex hull with their Delaunay triangle counterparts (same as for α-shapes), successively ordered by diminishing edge length. While this still guarantees the resulting polygon to remain manifold, it can

only reconstruct a single curve. The algorithm may also get stuck in case densely sampled points remain in its interior instead of being interpolated.

O'Rourke [OBW87] proposed to compute the minimal length tree in the Voronoi diagram corresponding to a polygonal boundary in its Delaunay triangulation dual, but this requires a distinct skeleton to work.

De Figueiredo and Gomes [FMG94] proved that the *Euclidean minimal spanning tree* (EMST) reconstructs open curves from sufficiently dense samples, with density defined by an empty tubular neighborhood.

Attali defined an r-regular shape [Att97] as having a boundary with curvature $\geq r$ everywhere. Then they proved its reconstruction from a uniform sampling of that boundary such that all discs with radius $< \frac{r}{2}$ centered on the boundary contain at least one sample.

Stelldinger [Ste08] proved both the correctness of the above-mentioned ball-pivoting algorithm [BMR*99] and a much more strict bound on required sampling of the points, but only for the uniformly sampled case: minimum 6 points for a sphere, as opposed to 22 [NSW08], 484 ($\varepsilon = 0.1$, [ABK98]) or 1343 ($\varepsilon = 0.06$, [ACDL00]).

Stelldinger and Tcherniavski [ST09] extended the proof above to noisy uniform samples.

Ohrhallinger and Mudur [OM11] also exploited the minimum length property of the EMST. They used edge exchange operations to transform it into a manifold. While they show successful reconstruction for several sparsely sampled and noisy point sets, there is no easily applicable sampling condition. Its time complexity is non-polynomial in principle, even if, for many cases, it terminates in $O(NlogN)$ time.

In a later paper [OM13], they define a modification of the EMST which relaxes its vertex valence constraint from ≥ 1 to ≥ 2. This *minimum boundary complex* can be approximated well in $O(nlogn)$ time. By inflating (a dual to the sculpturing [Boi84a] operation) they achieve a manifold boundary already close to the points. This facilitates the subsequent sculpturing step, with much-reduced risk of falling into local minima. They proved correct reconstruction for a tightened sampling condition of $\varepsilon < \frac{1}{2}$, although it additionally requires a local uniformity $u < 1.609$ as expressed in the proportion of the lengths of adjacent edges.

Peethambaran et al. [PM15b] defined a proximity graph called *shape hull graph (SHG)*, which faithfully reconstructs smooth curves that exhibit divergent concavity. The authors characterize the divergent concave curves based on the exterior medial balls in the pseudo-concave [PM15b] areas. The algorithm constructs the SHG by repeatedly removing boundary Delaunay edges subjected to geometric and topological properties. The geometric criterion (circumcenter location of the Delaunay triangle) is used to prune off elongated Delaunay triangles whose vertices lie further apart from each other on the curve, and the regularity criterion eliminates non-manifold elements, e.g., dangling edges and junction points, in the resultant polygon, thereby making it topologically equivalent to a circle (sphere in 3D).

In subsequent work, the authors [PPT*19] employed an incremental algorithm to classify Voronoi vertices into *inner* and *outer* with the help of normals estimated through Voronoi poles. Such a classification not only helps reconstructing the underlying curve but also aids in medial axis computation and dominant point detection. Theoretical guarantees under *bi-tangent neighborhood convergence*, a slightly modified version of divergent concavity for simple closed and planar curves, is also provided.

In a greedy approach introduced by Parakkat and Muthuganapathy [PM16], starting from the smallest edge in the Delaunay triangulation (which is guaranteed to be part of the reconstructed curve under ε-sampling), the algorithm iteratively adds an appropriate shortest edge to the result until it satisfies some conditions. The procedure is repeated to facilitate the capturing of disconnected components. Also, they employ a heuristic to identify whether the reconstructed curve is open or not.

Graph-based curve reconstruction methods often require the user to choose a global parameter, and in consequence, they yield good results only for uniformly sampled points. This means that for samples spaced too widely apart, the reconstructed curve may contain holes. On the other hand, if the spacing is too dense, not all samples may be interpolated by the output. On top of that, there is no guarantee that a Delaunay graph contains a polygon interpolating all samples.

3.2. Feature Size Criteria Reconstruction

Amenta et al. proposed in their seminal paper [ABE98] to apply the concept of the *local feature size* [Rup93] to the spacing of samples and define the *Crust* as a subset of the Delaunay triangulation of the point set. The CRUST algorithm which reconstructs this curve does not require the user to tune a global parameter for the (uniform) sample spacing. Instead, it permits reconstruction of this subset from non-uniformly sampled points, which is a curve as long as they conform to their stated sampling condition. This sampling condition requires a minimum angle between adjacent edges (assuming equal edge lengths) of the reconstructed piece-wise curve, which increases with their proportion of lengths. CRUST requires an ε-sampling of $\varepsilon < 0.252$, which corresponds to an angle $\alpha > 151.05^\circ$ between adjacent edges. While it is important as a theoretical result, in practice, these angle requirements are quite restrictive and difficult to ensure for point sets.

Gold [Gol99] developed a one-step algorithm that extracts above *Crust* without having to construct the Voronoi diagram on top of the Delaunay triangulation, and with it, the *Anti-Crust*, the skeleton approximating the medial axis.

Dey and Kumar [DK99] improved on that result with the elegant and simple NN-CRUST algorithm that relaxes the sampling condition to $\varepsilon < \frac{1}{3}$, corresponding to $\alpha > 141.62^\circ$. It first connects the points to their nearest neighbors and then adds a second edge, where necessary, to the nearest point such that it creates an angle $> 90^\circ$.

Dey et al. [DMR99] extended CRUST as well to CONSERVATIVE CRUST, which filters specific edges from the Gabriel graph. It can reconstruct (collections of closed and) open curves, and it is also robust to outliers. However, it requires a parameter and misses some sharp corners, which can be reconstructed by CRUST and NN-CRUST.

Lenz [Len06] claimes to relax the required density of NN-CRUST to $\varepsilon < 0.4$ and up to $\varepsilon < 0.48$ depending on the angle α but without proof. The proposed algorithm also permits the reconstruction of sharp corners and self-intersecting curves, starting with a seed edge between the two closest points and connecting edges by tracing along with a probe shape.

Hiyoshi [Hiy09] adapted the Traveling Salesman Problem to multiply connected curves, making it solvable in polynomial time as a maximum-weight 2-factor problem. The algorithm operates on the Delaunay triangulation and proved correct reconstruction for $\varepsilon < \frac{1}{3}$ and relative uniformity of adjacent edge lengths, differing at most by a factor of 1.4656.

Ohrhallinger et al. [OMW16] described a simple variant of NN-CRUST, which they call HNN-CRUST since they connect both nearest neighbor and a so-called *half neighbor* per point (unless it forms an endpoint of the curve). This half neighbor is defined as the nearest point lying in the half-space opposite the bisecting edge of the nearest neighbor edge. Connecting these neighbor points reduces the minimum angle from 90° (for NN-CRUST) to 60°. They improve the sampling condition for this algorithm up to $\varepsilon < 0.47$. Furthermore, they introduce a new reach-based sampling condition which they relate to ε-sampling, $\rho = \frac{\varepsilon}{1-\varepsilon}$. It manages to reduce the number of required samples for reconstruction and permit sharp angles by defining the distance to the medial axis at intervals between samples instead of at samples only.

3.3. Fitting Curves to Noisy Points

The algorithms mentioned above do not reconstruct curves well if the samples are contaminated by noise.

Lee [Lee00a] uses a technique called *moving least-squares (MLS)* [Lev98] which iteratively projects points on a curve fitting their local neighborhood by distance-weighted regression. This results in a thinned point cloud which can be locally approximated by a line inside a constant-sized neighborhood so that the center can be connected with its furthest neighbor points to form the edges of the reconstruction. The weighting function for the MLS projection considers points only inside a globally constant radius, which could also include unwanted points. Therefore the connectivity of points is created using the EMST, which minimizes edge length, and is then traversed to determine the local noise extent.

A noise-robust extension [MTSM10] of the *Hidden Point Removal (HPR)* operator [KTB07] computes local connectivity between points based on a projection onto their convex hull. The global reconstruction is then extracted by approximating the maximum weight cycle from a weighted graph combining the local connectivity. The algorithm does not denoise or smoothen, i.e., simply interpolates points, and the reconstruction exhibits holes or misses points in regions with moderate noise extent.

Rupniewski [Rup14] first sub-samples a noisy point set with minimum (globally constant) density. Then he uses a heuristic that alternatingly moves these points to local centers of mass based on the Voronoi diagram and eliminates points that do not have exactly two neighbors in a density-sized neighborhood until the point set is stable. Finally, after hundreds of iterations, the points can be ordered consecutively. Only very basic results are shown in the paper.

Cheng et al. [CFG*05] resample a thinned point set from noisy points and then use NN-CRUST to reconstruct the curve. They prove a probabilistic sampling condition, however, it is impractical due to its restrictive sampling density constraints, and they only prove but do not show any results of their proposed algorithm.

Wang et al. [WYZ*14] first construct a quad-tree on the samples to determine the inner and outer boundaries of noisy samples on a grid. After smoothing these boundaries, they compute their Voronoi diagrams in order to extract the skeleton which represents the reconstructed curve. While their method is very resilient to outliers and noise, it requires careful tuning of several parameters and does not handle sparse samples well.

FITCONNECT [OW18a] seamlessly extends parameter-free HNN-CRUST to handling noisy samples. The conforming condition of the latter, which specifies whether three points can be connected in exactly a single way is extended by fitting a circular arc to the local neighborhood if consisting of more than three points. Where local fits do not overlap consistently, they are grown to larger neighborhoods until covering these noisy clusters. The resulting ordered consistent local fits are then denoised to this locally estimated noise extent (the variance of the fits) by blending them together. The algorithm also manages to classify sharp corners which would otherwise be smoothed. Its runtime is however, $O(k^2)$ in the size k of noisy neighborhoods.

STRETCHDENOISE [OW18b] improves the blending technique used for denoising in FITCONNECT by modeling the recovered manifold connectivity separated from the high-frequency residuals. These are used to shift point positions by minimizing angles between edges in the least-squares sense. Additionally, the movement of points is restricted to lie inside a probability density function cut-off distance, which is estimated from the variance of the fitted arcs but can also be input from sensor noise models. This also guarantees stochastic error bounds for the noisy samples.

Fitting curves to approximate noisy samples is a difficult task and trades off recovering feature detail vs. robustness.

It is worth noting that, for some applications, instead of a polygonal reconstruction, they prefer the reconstructed result to be a polynomial curve(s). A few among such applications include vector representation of computer fonts [IG16] and reconstructing the contours of medical images [IGA18]. Unlike polygonal reconstruction methods (which is the main focus of this report - and uses straight lines to connect appropriate points), these methods fit the input points by polynomial curves (mainly B-Splines or Bezier curves) that minimize a particular cost function.

3.4. Reconstructing Sharp Features

The well-known and established ε-sampling condition has a significant drawback; it cannot sample a sharp corner. That is because the medial axis touches the corner and hence would require an infinite amount of samples at that particular point to satisfy ε-sampling for any ε.

Assuming a new sampling condition based on the tangential circles with respect to a point in the curve (to avoid the need for infinite sampling at sharp corners), GATHAN [DW01] modifies the nearest

neighbor strategy to handle sharp corners. While selecting an edge e, the improved algorithm takes into account both the angle between the dual Voronoi edge and the estimated normal of e, ratio of its dual Voronoi edge length to its length and the degree of all the vertices. This method is, later on, extended [DW02] by carefully structuring it to provide a theoretical guarantee. The improved algorithm requires only one parameter, which gives the minimum angle of all sharp corners based on which the sharp corners are locally sampled.

Rather than imposing extra conditions, Funke et al. [FR01] proposed an algorithm that guarantees to reconstruct the curve faithfully under a specific sampling condition. They proposed a sampling condition relying on the edges of the correct reconstruction for a smooth curve and later on relax it at corner points to generate a weak sampling around it. Starting from justifiably 'smooth' edges, their reconstruction explores potential corners. The identified corner edges are then merged with the smooth edges to give the final reconstructed result.

While algorithms specialized to handle sharp features reconstruct these cases quite well, their conditions are often complex, and they do not compete well for the general case.

3.5. Traveling Salesman Methods

Giesen [Gie99] showed in an existence guarantee that for sufficiently dense sampling, the boundary can be reconstructed by solving the *Euclidean Traveling Salesman Problem* (ETSP) for a set of points. He proposes two algorithms in that paper but does not present any results.

Althaus et al. [AM00] add to this that it also works for non-uniform sampling. Furthermore, they show that if constrained by an ε-sampling [ABE98], the NP-hard ETSP terminates in polynomial time. However, they manage to prove that only for a very restrictive $\varepsilon < \frac{1}{20}$, which permits just angles $> 174.27°$.

Althaus et al. [AMS00] compared approximation algorithms for the ETSP, based on heuristics, but noted that these all fail for sparsely sampled cases where the ETSP would have succeeded. An interesting observation is that the complexity for the ETSP construction decreases as the sampling gets denser. A naive ETSP construction takes $O(2^n)$ time which is unfeasible even for very small point sets.

Arora et al. [Aro98] approximate the ETSP within $(1+\frac{1}{c})$ in $O(n(logn)^{O(c)})$ time, but their reconstructions result in poor visual quality. The fastest exact TSP solver, the *Concorde* [ABCC], would still take years to compute the boundary for practical point sets, as can be derived from a discussion of its complexity [HS09].

In analogy to the Traveling Salesman problem of minimizing curve length, polyhedra with minimal area were proposed for surface reconstruction [O'R81]. But for this NP-hard problem no algorithm exists. Furthermore, Boissonat showed by a simple example that this minimum is not always visually pleasing [Boi84b].

The runtime performance of the ETSP degrades drastically for sparsely sampled points. Those point configurations tend to be rather ambiguous w.r.t. which points are connected to the boundary. Therefore, the ETSP is not a suitable tool for curve reconstruction as it puts almost all computational effort where it makes little difference in terms of visual aesthetics.

3.6. Curves with Self-Intersections

Most of the curve reconstruction work concentrates on reconstructing a (set of) simple closed or open curve(s). But for applications like sketching [PMM18] or point sets generated from images, the inputs might also contain self-intersections.

The first one in this category [DGCSAD11] formulates and solves an optimal transport problem. Starting from the Delaunay triangulation of the input point set, their approach generates a coarse mesh in a greedy fashion with the objective of minimizing the total cost. With the help of intelligent vertex relocation, this approach is specially designed to handle noise and outliers. Since this procedure does not impose any manifold or degree constraints on the input while filtering out edges from the simplified mesh, it can reconstruct shapes with self-intersections.

Instead of modifying the algorithm to adapt to reconstructing curves with self-intersections, Parakkat et al. [PMM18] use a post-processing step to identify and restore self-intersections. Initially, a reconstruction step with a vertex degree constraint of a maximum of three is used. Later on, potential intersections are explored at the vertices with degree one. Based on a user parameter and the one-ring Delaunay neighborhood of the considered vertex, potential self-intersections are recovered by the appropriate Delaunay edges.

It is worth mentioning that even if CRUST [ABE98] and NN-CRUST [DK99] are not particularly designed to handle curves with intersections, in some cases, they capture self-intersections since they do not impose a manifold restriction on the vertex degree.

3.7. Curve Generation based on Human Visual System

A few curve reconstruction algorithms rely on a subset of Gestalt laws of perception, which describe how humans perceive visual elements. Among the six Gestalt rules, *proximity* and *continuation* are very important to curve reconstruction strategies. While the proximity rule suggests that the human visual system has a natural tendency to group nearest points, the law of continuation states that the human eyes will follow the smoothest path when viewing curves and hence helps to guide our eyes in a certain direction while connecting the points [Mat16]. Following the Gestalt law of proximity, Zeng et al. [ZNYL08] proposed a parameter-free algorithm called DISCUR for reconstructing multiple simple curves that may be closed or open as well as contain sharp features. A successful reconstruction using DISCUR depends on an appropriate sampling of interior curves and an accurate identification of boundary curves. Since DISCUR relies on the proximity criterion, wrong connections may occur when a sample has two or more nearest neighbors. In such cases, the selection is quite arbitrary. Furthermore, it requires a very dense sampling near sharp corners in order to reconstruct these correctly.

An improved version of DISCUR has been presented as well [NZ08]. The authors utilize the Gestalt principles of proximity and continuity to formulate a vision function that is supposed to best mimic the natural human vision. The main intuition behind the algorithm is that any abrupt changes while connecting the points are reflected in the statistical properties of the curve, which are in turn captured through the vision function. The algorithm, known as VICUR, employs appropriate rules based on the vision function to

reconstruct multiple closed or open curves with or without sharp features. A drawback of VICUR algorithm is that it is highly sensitive to the user-tuned parameters.

Note that Edge exchanging [OM11] and Connect2D [OM13] algorithms (already described in Subsection 3.1 both also relate to the Gestalt laws.

4. Implicit Curve Fitting

Implicit functions for curve or surface fitting have been widely investigated in the computer graphics community. Implicit methods attempt to define a smooth function $f\colon\ \mathbb{R}^2 \to \mathbb{R}$ such that the zero level set of f approximates the underlying curve in the input points as illustrated in Figure 2. The zero level set of the curve (also referred to as contour) or surface can be directly visualized by using a ray tracer or by polygonizing it using the well-known marching squares or cubes algorithm [LC87]. Many algorithms have been developed for algebraic curve fitting to a set of 2D or 3D points, e.g., conic planar curves [Boo79, FMZB91] and curves of arbitrary degrees [Pra87, Tau91, Tau93]. A curve or surface is *algebraic* if their representative functions, i.e., f are polynomials of some degree d. Fitting algebraic curves to a finite set of points is normally posed as a least-square fitting problem where the objective is to minimize the mean square distance from the sample points to the curve.

In general, implicit functions are extremely compact and suitable for representing free-form curves [dALJ*15]. Though most of the implicit techniques focus on surface fitting, many of them can be either directly applied or adapted for planar curve fitting. A typical choice for the implicit function is the *Signed distance function (SDF)*. The SDF for an arbitrary point p is the signed distance between p and its nearest point on the boundary where the sign component indicates the location of p with respect to the curve, i.e, whether the point lies inside or outside the boundary. Reconstruction methods employing SDF range from tangent plane estimation [HDD*92] to polynomial splines over hierarchical T-meshes [SJP10].

Radial basis functions (RBF) represent an excellent tool for the smooth interpolation of scattered points. Carr et al. [CBC*01] formulate the implicit function f as a linear sum of weighted and shifted radial functions, i.e., $f(p) = \sum_{i=1}^{n} w_i \phi(\|p - c_i\|)$, where the weights w_i are determined by solving a linear system constructed from various surface constraints at input points c_i. Choices for the basic function ϕ include Gaussian ($\phi(r) = exp(-cr^2)$), multi-quadric ($\phi(r) = \sqrt{c^2 + r^2}$), polyharmonic ($\phi(r) = r$ or $\phi(r) = r^2$) and thin-plate spline ($\phi(r) = r^2 \log r$). In a related work [TO02], the authors estimate the implicit surface as an RBF that minimizes thin-plate energy subject to a set of interior and exterior constraints.

Poisson reconstruction [KBH06] solves for an indicator function for the curve (or surface), whose gradient best approximates the normal field N, i.e., $F = \text{argmin}_S \parallel \nabla_S - N \parallel_2^2$. This optimization problem leads to a Poisson equation, which is solved by a locally supported radial basis function on an adaptive octree (quadtree in 2D). As the current gold standard in the community for surface reconstruction, it however, requires normals at points to be specified. In related works, Fourier [Kaz05] and wavelet [MPS08] bases have been employed for an accelerated solving of Poisson equations.

An alternative to RBF curve interpolation is the moving least square (MLS) method. The MLS projection [Lev04] method first defines a local reference frame $\mathcal{H}$ for a point q to be projected and then fits a local polynomial approximation g to the weighted input points. Here, the weight for each input sample p_i is a function of its distance to the projected q on $\mathcal{H}$. Once the polynomial is computed, the projection of q onto g represents the MLS projection of q. Lee [Lee00b] mentioned above describes an improved moving least-squares technique using Euclidean minimum spanning tree and region expansion for fitting non-intersecting curves to unorganized point clouds. Several MLS based approaches including surface resampling [ABCo*03], progressive point set surfaces [FCOAS03], sharp feature reconstruction [FCOS05], algebraic spheres (or circles) [GG07], provable MLS surfaces [Kol08], have been proposed. Being insensitive to noise, MLS approaches are suitable for fitting curves to noisy data.

The idea of decomposing the input data domain into sub-domains and locally fitting piece-wise quadratic functions to the data is prevalent in the surface reconstruction and is equally applicable to 2D curve fitting. Ohtake et al. [OBA*03, OBS06] blend the locally fitted quadratic functions using a weighing function (the partitions of unity) to create the global approximation to the underlying surface. Alliez et al. [ACSTD07] utilize a Voronoi diagram of the input point set to deduce a tensor field whose principal axes and eccentricities locally represent the most likely direction of the normal to the surface and the confidence in this direction estimation respectively. An implicit function is then computed by solving a generalized eigenvalue problem such that its gradient is most aligned with the principal axes of the tensor field, providing a best-fitting iso-surface or curve reconstruction.

In general, implicit curves employ acquired or estimated point normals to facilitate the reconstruction process and are found to be robust against noise. However, since the iso-curves are extracted using marching squares on quadtrees, an appropriate quadtree depth has to be determined and preset in the case of implicit methods. Detailed curves can be generated for larger depth values, however, at the expense of increased computational time. While popular in surface reconstruction, we have not found implementations or results for curve reconstruction, and so this category is excluded from our comparison or evaluation.

5. The Benchmark

In this section, we briefly describe the curve reconstruction benchmark. The motivation behind setting up such a benchmark is to encourage the use of standardized datasets and evaluation criteria for research and advancements in curve reconstruction and related applications. The proposed benchmark repository consists of a driver program, data sets, associated ground truth, sampling, and evaluation tools in the shape of test scripts for a comprehensive experiment on 2D curve reconstruction algorithms. We also provide a set of publicly available curve reconstruction algorithms in the benchmark. The selected algorithms include early ones from the late nineties up to recent papers. The components of the benchmark repository and their interactions are illustrated in Figure 1 and discussed in the following sections (Sections 5.1-5.5). Full source of the benchmark is available here: `https://gitlab.com/stefango74/curve-benchmark/`.

5.1. Algorithms

We have included a set of fifteen publicly available curve reconstruction algorithms in the benchmark. Table 2 records the list of algorithms and the abbreviations that we use in our experiments to refer to them. Note that we were not able to obtain code for some of the algorithms ([MTSM10], [Lee00a], [WYZ*14], [Hiy09]) and therefore could not include those in the benchmark. All the algorithms except OPTIMAL TRANSPORT [DGCSAD11] interpolate or try to closely fit the input points. On the contrary, OPTIMAL TRANSPORT focuses on simplified reconstruction, and hence, for a fair comparison, we have not included it in any of our experiments but presented some representative results.

Algorithm	Open Source
CRUST [ABE98]	yes
NNCRUST [DK99]	yes
CCRUST [DMR99]	yes
GATHAN [DW01]	yes
GATHANG [DW02]	yes
LENZ [Len06]	yes
CONNECT2D [OM13]	yes [Ohr13]
CRAWL [PM16]	yes [Par16]
HNNCRUST [OMW16]	yes [Ohr16]
FITCONNECT [OW18a]	yes [Ohr18a]
STRETCHDENOISE [OW18b]	yes [Ohr18b]
PEEL [PMM18]	yes [Par18]
OPTIMALTRANSPORT [DGCSAD11]	yes [ACSd*18]
DISCUR [ZNYL08]	yes
VICUR [NZ08]	yes

Table 2: *Algorithms compared in our study. The source is provided together with our benchmark unless referenced here, in which case it will be pulled from the respective repository.*

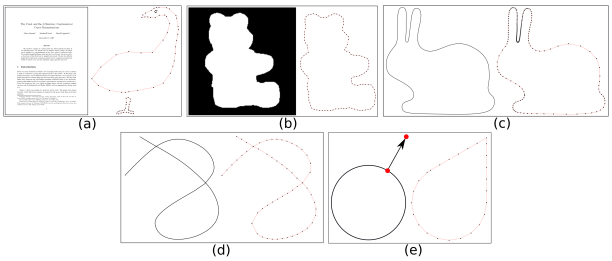

Figure 7: *Examples of different types of test data. (a) Classic data collected from different papers, (b) Points sampled from a binary image boundary, (c) LFS-sampling from a cubic Bézier curve, (d) Points sampled from a synthetic curve, (e) Synthetic data generated by extruding sharp corners from circles.*

5.2. Data Sets and Associated Ground Truth

We collected data from various sources as well as synthetically generated test cases using analytical functions. We also provide the ground truth associated with the test data if they represent a linear approximation to the input curve. We classify the test data based on the data source and mode of generation (see Figure 7 for examples from the different categories) as follows:

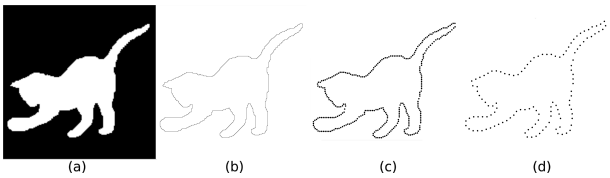

Figure 8: *(a) A binary image, (b) Extracted edges, (c) Result of sampling with radius = 20, (d) Same for radius = 50.*

CLASSIC This data set consists of all the point sets collected from various curve reconstruction papers or projects, mostly from the project page or repository. A few data sets were extracted from the images using webplot digitizer [Roh20]. Test data include well-known point sets used for evaluating the reconstruction quality of sharp features, open curves, and sharp corners. We collected 25 manifold curves, 16 non-manifold curves, 21 curves with sharp corners, 23 open curves and 52 multiply connected curves.

IMAGE This set consists of contours extracted from silhouette images (see Figure 8) from various image databases, e.g., MPEG-7 Core Experiment CE-Shape-1 Test Set [mpe02], Edinburgh Kitchen Utensil Database [AD15], and the 1070-Shape Database [107]. A description of how the contours (2158 manifold, 2 multiply connected, 206 sharp corners) are extracted is given in Subsection 5.3.

SYNTHETIC Two analytical shapes (bunny, sharp corner) were sampled with this method [OMW16], also detailed in Subsection 5.3. These data sets were used for our experiments on feature-sized noise, sampling density, and curves with sharp features. The code to generate sharp curves with varying degrees of sharpness (see Figures 7(d)- 7(f)) is available as a part of *CurveBenchmark.cpp*, the driver program.

All the experimental data are organized inside different subdirectories, i.e., *multiple-curves, open-curves, sharp-corners, non-manifolds*, and *manifolds*. Test data sampled from curves exhibiting multiple features are placed under more than one directory. Each test case that we experimented with has an associated ground truth which is used to evaluate the reconstruction quality. This ground truth is represented using either an *indexed list* or *ordered vertices*. In the indexed list representation, the ground truth file stores all the vertices first, followed by the pairs of vertex indices representing the edges. Ordered vertices represent the edges of the curve using consecutive vertices in the file.

5.3. Sampling Tools

To analyze how reconstruction algorithms perform w.r.t. varying sampling density, we repeat here a simple sampling algorithm [OMW16] that creates an approximate ε-sampling on cubic Bézier curve input.

First, we densely sample the segments of the Bézier curve along its parametrization. Then the normal n_i at each curve sample $s_i \in S$ is computed as orthogonal to the edge connecting its neighbor samples on the curve. The largest empty disc at s_i can be established by

s_i, n_i and querying each other curve sample $s_j \in \{S \setminus s_i\}$ by setting the disc center $c_j = s_i + tn_i, \|cs_i\| = \|cs_j\|$. After solving this, the c_j with the largest radius of all empty discs are added to the set of medial axis points M. Now, having sampled this medial axis approximation, we can simply estimate the lfs for each s_i by locating its nearest neighbor in M and its distance. Note that this computation is not exact due to the discretization of the original curve as well as floating-point precision error. However, computing medial axis and thus the lfs exactly is a hard and computationally expensive task [AAA*09], [ABE09]. But since ε-sampling requires an upper bound on distance, and the curve is also discretized, the chosen samples should be mostly within that bound. In order to sample the curve with a given ε, we now start with any curve sample s_i (or any on its boundary if the curve is open) and iterate over successive samples along the curve while $\|s_i, s_j\|/\text{lfs} < \varepsilon$ and choose the last valid one as next point in our ε-sampling.

Since we have now computed the lfs for all samples, we can further perturb these in relation to the lfs, in order to simulate feature size varying noise. We retain the sampling density by just moving each sample along their normal, which was incidentally determined by the fitting of the empty discs.

We also provide a discrete sampling tool written in Processing3 (`www.processing.org/`) for extracting points from a given binary white-on-black image. It first extracts the pixels lying on the object boundary by comparing each pixel with its 8-neighborhood. Then, the extracted boundary is fed to a boundary sampling algorithm. Based on a user given radius r (which determines the sampling density), the sampling algorithm randomly picks a pixel at position (x,y), inserts a point at the location (x,y), and erase all boundary pixels lying at a distance less than r from (x,y). This procedure is repeated until all boundary pixels have been erased. Figure 8 shows a sample binary image, its extracted edges, and samples generated for two radius values.

Interactive sampling: For some randomly sampled point sets, identifying the ground truth is tricky since no algorithm can claim a proven reconstruction. In such cases, in order to help the user generate the ground truth, we use an interactive ordering program. The interactive ordering program displays the point set, the user selects points by clicking on them, and the points are saved in this user-specified order. This program then also displays the edges according to that order.

5.4. Evaluation Criteria

In order to measure how well a reconstructed curve C' approximates the original C, we compute the distance for closest points between the two curves, similar to this benchmark [BLN*13] between both shortest distance maps $M : C \mapsto C'$ and $M' : C' \mapsto C$ since the mapping is not bijective: $M' \neq M^{-1}$.

We sample sets of points S, S' on the two respective curves C, C' (which consist of edge-chains) uniformly and densely. Then for each sample $s \in S$, we determine its closest point $t \in C'$ to create a discrete mapping (s,t) from all $s' \in S'$ to $t' \in C$, and a similar reverse mapping (s',t').

The set of closest point correspondences are then:

$$D = (s,t) | s \in C', t = M(s) \tag{1}$$

$$D' = (s',t') | s' \in C, t' = M'(s') \tag{2}$$

Based on these mappings, with $N = |D| + |D'|$, we can approximate the following metrics, first Hausdorff distance:

$$H_D(C,C') = \max \left\{ \max_{(s,t)\in D} \|s-t\|, \max_{(s',t')\in D'} \|s'-t'\| \right\} \tag{3}$$

and then root mean squared distance:

$$RMS_D(C,C') = \sqrt{\frac{1}{N}\left(\sum_{(s,t)\in D} \|s-t\|^2 + \sum_{(s',t')\in D'} \|s'-t'\|^2 \right)} \tag{4}$$

Note that we do not evaluate distance bilaterally as our sampling algorithm requires manifold closed curves, whereas the reconstructed curves in our experiments may be open and non-manifold, and therefore measuring the distance from a sampling on them would be less meaningful.

5.5. Benchmark Driver and Test Scripts

CurveBenchmark.cpp is the C++ driver program of the curve reconstruction benchmark. This program consists of an algorithm list and functions for input-output processing, sampling, noise or outlier synthesis and evaluation. The driver program can be run from the terminal or using a test script. While running the driver executable, necessary arguments along with the command-line options should be provided to interpret the argument type or values. A few examples of arguments and associated options are input file ($-i$), output file ($-o$), algorithm name ($-a$), and ground truth file ($-g$). The parameter $-h$ displays the list of options and their usage.

The architecture of our benchmarks consists of some test scripts that quantitatively and qualitatively evaluate the curve reconstruction algorithms by feeding the input data sets to algorithms and processing the evaluated data into graphs. Each test script includes a list of algorithms that need to be considered for the experiment and a list of test data. To add a new reconstruction algorithm, namely A, for a particular curve feature, A has to be first downloaded to the benchmark repository, compiled, and the executable of A has to be linked to the benchmark driver via the given make file. Additionally, A needs to be included in the algorithm list of the benchmark driver program. Finally, A has to be included in the appropriate test script and to be run, which in turn invokes the benchmark driver, thereby generating the resulting polygonal curves as well as the graph plots. All this can simply be copied and modified from the existing structure. It should be noted that our test scripts also make it easy for the benchmark users to run selective experiments using a subset of the benchmark data. The list of test scripts in our benchmark evaluates RMS error for the following input data unless otherwise noted:

- **run-sampling.sh:** ε-sampled [ABE98] test data
- **run-noisy.sh:** perturbed with uniform noise
- **run-lfsnoise.sh:** perturbed with lfs-based noise
- **run-outliers.sh:** added outlier points
- **run-manifold.sh:** whether reconstruction is a manifold
- **run-sharp-corners.sh:** sharp feature curves
- **run-open-curves.sh:** open curves
- **run-multiple-curves.sh:** multiply connected curves
- **run-intersecting.sh:** curves with intersections

6. Evaluation & Results

In this section, we demonstrate the utility of the benchmark by comparing the curve reconstruction algorithms included in the benchmark. The algorithms have been evaluated on different test data using various criteria implemented in the benchmark. For the sake of fair comparison, a majority of the experiments use only the interpolatory algorithms, i.e., optimal transport [DGCSAD11] has been discussed separately. Both the quantitative and qualitative comparisons have been presented along with a detailed interpretation of the results. Besides the comparison of classic and recent algorithms, this section also focuses on the demonstration of a comprehensive experimental design for curve reconstruction techniques, i.e., to show how the quantitative and qualitative comparisons are designed and performed using the test data and the scripts available in the curve reconstruction benchmark. Note that any new curve reconstruction algorithm can be easily added to the benchmark by duplicating and adapting the appropriate wrappers and subsequently be compared with the existing algorithms in the benchmark.

6.1. Quantitative Evaluation

We rank the 14 algorithms by the following six aspects. The closeness to the original is measured by computing the root of the mean-squared error (RMSE) metric. This permits determining the robustness with respect to various sampling artifacts.

- Sampling density as ε-sampling
- Noise robustness as δ of bounding box diagonal
- Noise robustness as δ of lfs
- Noise+sampling density as ε-sampling and δ of lfs
- Outliers robustness in % of samples
- Average runtimes (in s)

The test set consists of point sets sampled from multiply connected and disconnected curves and curves with sharp corners. For some experiments, we used smooth curves of the bunny sampled as required (see Figures 9, 11 and 12).

6.1.1. Sampling Density

First, we look at simple reconstruction from a non-uniform sampling of curves, free of artifacts such as noise or outliers. For this, we determine an ε-sampling on a cubic Bézier curve (bunny), see Subsection 5.3 for our detailed implementation. Reconstruction from sufficiently dense samples is not a difficult task. In order to show which algorithms also work well on sparser sampling, we sample with decreasing density: $\varepsilon = 0.25, 0.5, 0.75$ (see Figure 9). As can be seen, irrespective of the density, for dense sampling (ε = 0.25,0.5),

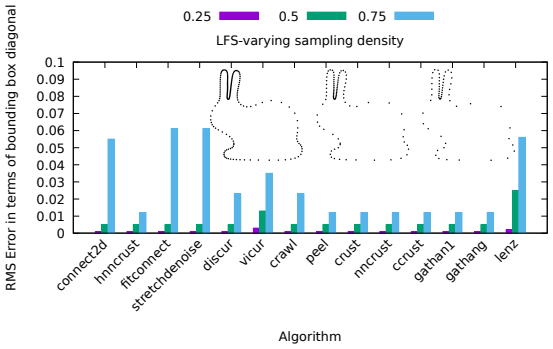

Figure 9: *RMS Error of reconstructed curves from ground truth for a cubic Bézier curve sampled with* $\varepsilon = 0.25, 0.5$ *and* 0.75 *(*run-sampling.sh*). The point sets sampled from the bunny curve are shown in the figure.*

all the algorithms perform equally well (except for LENZ - which gives a comparatively poor performance in all cases, and VICUR). But, as the sampling becomes sparse (ε = 0.75), the performance of algorithms like CONNECT2D, FITCONNECT, and STRETCHDENOISE deterioratess compared to other algorithms. Note that minimum error levels stem from comparing the coarser reconstruction to the original smooth curve.

6.1.2. Robustness to Noise

To evaluate the performance of the algorithms on the noisy point set, we ran our benchmark on point sets perturbed by different noise levels. In our first experiment, we introduced uniform noise, then, we varied the noise in terms of the local feature size, and finally, we fixed the latter while varying the sampling density (also in terms of the lfs):

Uniform Noise:

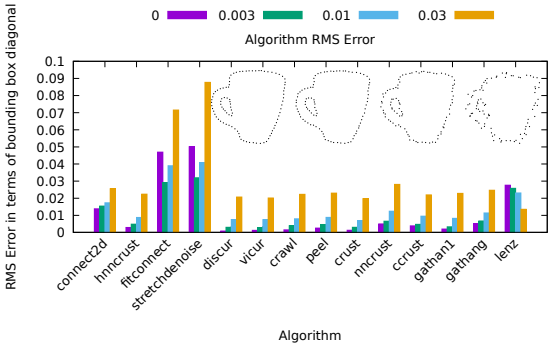

Figure 10: *RMS Error of reconstructed curves from ground truth for the point sets used in Figure 16, with the points perturbed with* uniform *noise of* $\delta = 0.003, 0.01$ *and* 0.03 *as well as the non-noisy original (*run-noisy.sh*). An example input with varying noise is also shown in figure.* LENZ *results are not directly comparable as it reconstructs a non-manifold and thus largely the original as a subset.*

We simulated noise of maximum extent δ by defining perturbing

a sample on the curve s_i to $s_i' = s_i + \sigma v_i$, where σ is an uniform random variable in $[0, \delta]$, δ is in terms of the bounding box diagonal, and v_i is a unit vector of uniformly random direction, similar to the noise model used here [MTSM10].

Using different perturbation levels of 0.003, 0.01, 0.03, we generated 3 additional noisy datasets from the original for the 25 CLASSIC-*manifold* points sets. Figure 10 shows a graph representing the performance of the various algorithms on our synthetically generated noisy dataset together with the non-noisy original. For the latter, reconstruction often fails already. The close plots show how it degrades with additional noise, except for LENZ, which reconstructs less wrong cross-connections. HVS- and Delaunay-based algorithms such as CRAWL, PEEL, CRUST families gave the best performance for uniformly distributed noise, while CONNECT2D, FITCONNECT, and STRETCHDENOISE fail to reconstruct the original in several cases due to sharp corners or too sparse sampling, but degrade much less for correctly reconstructed ones.

LFS-based Noise:

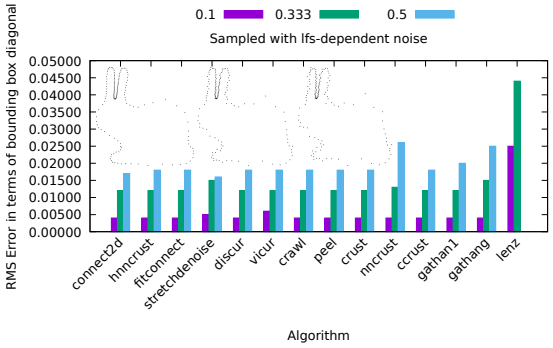

Figure 11: *RMS Error of reconstructed curves from ground truth for the sets of cubic Bézier curves in Figure 9, sampled with* $\varepsilon = 0.3$*, and the points perturbed with* local feature sized *noise of* $\delta = 0.1, \frac{1}{3}$ *and* 0.5 *(*run-lfsnoise.sh*). The figure also shows the lfs sampled bunny curve with varying noise (*LENZ *crashes for* $\delta = 0.5$*, therefore the false 0 value).*

In this experiment, we simulate noise in terms of the extent of the *local feature size*. To add noise to such a sampling of the bunny as above, we perturb the sample s_i only along the curve unit normal vector n_i in order to preserve the sampling density as well as possible, such that $s_i' = s_i + \sigma n_i$, with uniform random $\sigma = [-\delta, +\delta]\text{lfs}(s)$. Figure 11 shows the performance of the various curve reconstruction algorithms compared to the noise-free original, all with an ε-sampling of $\varepsilon = 0.3$. The algorithms show similar characteristics as for global uniform noise, as shown above. Algorithms such as CONNECT2D, FITCONNECT and STRETCHDENOISE showed superior performance under the lfs-noise model, similar to the global uniform noise model tested above. This is no surprise as the algorithms such as FITCONNECT and STRETCHDENOISE are specially designed to handle noise and provide guarantees under the *lfs* based sampling.

Varying Sampling Density + LFS Noise:

Now we introduce noise as above but with fixed $\delta = \frac{1}{3}$ and vary

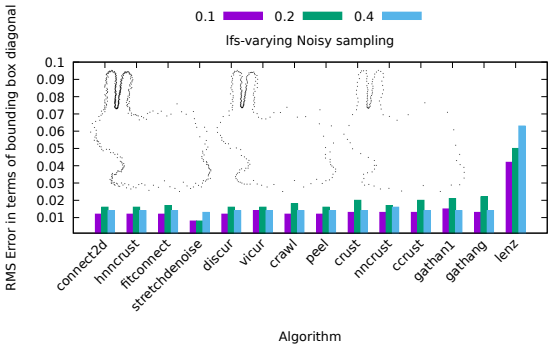

Figure 12: *RMS Error of reconstructed curves from ground truth for sets of cubic Bézier curves, with the points perturbed with noise of* $\delta = \frac{1}{3}$ *and sampled with* $\varepsilon = 0.1, 0.2$ *and* 0.4 *(*run-sampling-noise.sh*). The point sets of the bunny in the figure represent the inputs with varying sampling and noise.*

the sampling density with $\varepsilon = 0.1, 0.2, 0.4$ on the bunny, again according to Subsection 5.3. We can see in Figure 12 that varying sampling density does not have a large impact on reconstruction from noisy samples across all algorithms.

6.1.3. Outliers

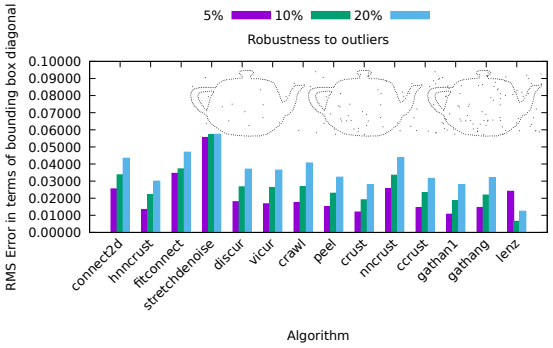

Figure 13: *RMS Error of closed reconstructed curves for the classic point sets used in Figure 16, with a share of* 5%, 10% *and* 20% *outliers added. (*run-outliers.sh*).* LENZ *results are not directly comparable as it reconstructs a non-manifold and thus largely the original, while connecting outliers.*

To evaluate the performance of reconstruction in the presence of outliers, we simply generate $n\%$ additional samples uniformly distributed inside the bounding box of the point set. All the interpolatory algorithms were tested on 25 input curves from CLASSIC dataset. Figure 13 shows the robustness of reconstruction after adding a varying amount of outliers as uniformly random distributed points inside the bounding box of the input point set. While CRUST and HNNCRUST algorithms perform very well in the presence of outliers, other algorithms such as STRETCHDENOISE and FITCONNECT are found to be comparatively sensitive to outliers.

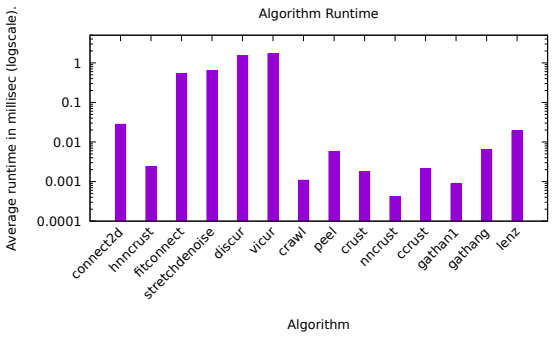

Figure 14: *Averaged runtime for the reconstruction of 2183 manifold point sets for 14 algorithms (*run-manifold.sh*). Note the log scale due to the large variance.*

Algorithm	Runtime(s)
CRUST [ABE98]	0.0018
NN-CRUST [DK99]	0.0004
CCRUST [DMR99]	0.0022
GATHAN1 [DW01]	0.0009
GATHANG [DW02]	0.0066
CONNECT2D [OM13]	0.0277
CRAWL [PM16]	0.0012
HNN-CRUST [OMW16]	0.0023
FITCONNECT [OW18a]	0.5081
STRETCHDENOISE [OW18b]	0.6182
PEEL [PMM18]	0.0058
DISCUR [ZNYL08]	1.5291
VICUR [NZ08]	1.7037
LENZ [Len06]	0.0576

Table 3: *Average runtime of 14 algorithms reported for 2183 manifold point sets.*

6.1.4. Computational Time

Figure 14 shows a bar chart of average run times for the various curve reconstruction algorithms. The test set consists of 2183 noise-free manifold point sets with an average of 283.9 points per test case, each of which represents a closed boundary of either a silhouette image or a geometric shape. We use CLASSIC as well as IMAGE contours for the experiment. Nearly 99% of the inputs were extracted from silhouette images available in different image databases mentioned in Section 5.2. The exact time values are also reported in Table 3. Computational times are based on the experiment performed on an Intel core-i7 2.4 GHz CPU with 16 GB memory. As indicated by the bar chart and the Table, NNCRUST is the fastest of all the compared algorithms with an average time of 0.0004 ms for reconstructing 2183 noise-free point sets, contrasting to 1.78s for the slowest, VICUR, with the majority of algorithms finishing in <10ms.

6.2. Qualitative Comparison

We compare how well the algorithms perform on these qualitative aspects:

- Manifold curves
- Non-manifold
- Sharp corners
- Open curves
- Multiple curves

Figure 15 shows example reconstruction results by the 11 algorithms which are able to handle all these aspects contained in the leaf input (sharp-corners, non-manifold edges, multiple and open curves).

6.2.1. Manifold Curves

We evaluate the reconstruction algorithms for their performance on the manifold curves. The set of 14 algorithms considered for this experiment were run on 2183 noise-free point sets used for the run time experiment (Section 6.1.4). Figure 16 shows the qualitative performance of the algorithms for exact reconstruction of manifold curves. The percentage of exactly reconstructed manifold curves lies in the range of 22.6-45% for the majority of algorithms. Among the tested algorithms, CONNECT2D performs best, with 52.13% of curves reconstructed faithfully. Contrary to the design objectives of LENZ, only 1.7% of the manifold curves were reconstructed correctly. This could be partly due to the presence of holes or multiply connected curves, as we observed that LENZ algorithm does not handle multiple curves well.

6.2.2. Non-manifold Point Sets

Figure 17 shows the bar chart of our experimentation on non-manifold curves. In this experiment, we have included only five algorithms that are known to handle non-manifold curves. We used 16 point sets created from various papers and ten synthetic point sets for the experiments. The synthetic point sets were generated by sampling randomly drawn self-intersecting curves. As can be seen, all the five algorithms gave a similar performance on non-manifold curves in terms of RMS error. In terms of the percentage of correct reconstruction, approximately 11% of the curves were faithfully reconstructed, which is the maximum by any of the tested algorithms. From the results, we observed that all the algorithms except LENZ reconstructed the curves reasonably well. However, the resulting curves lacked one or two edges and, therefore, did not exactly match the corresponding ground truths. This could potentially lead to such a low percentage of correct reconstruction.

6.2.3. Sharp Corners

Figure 18 shows the bar chart representing our experiments on curves with sharp corners. We tested algorithms that handle sharp corners on a test set consisting of 47 input curves. The test set consists of classic curves collected from the literature and synthetic data. Each curve in the synthetic dataset is a closed convex curve which was generated using an arc of a unit circle opening to two tangent lines joining at a specified angle to form a sharp corner. Circles were sampled with 10, 16 and 20 points with opening angles of the tangents of 10-90 degrees in steps of 10 and 5 and sampled with similar density. GATHANG performs best. It was specifically designed for curves with sharp features and correctly reconstructs 80.85% of the test cases. The second best algorithm in handling sharp corners is CONNECT2D, which handled 65.95% of the inputs

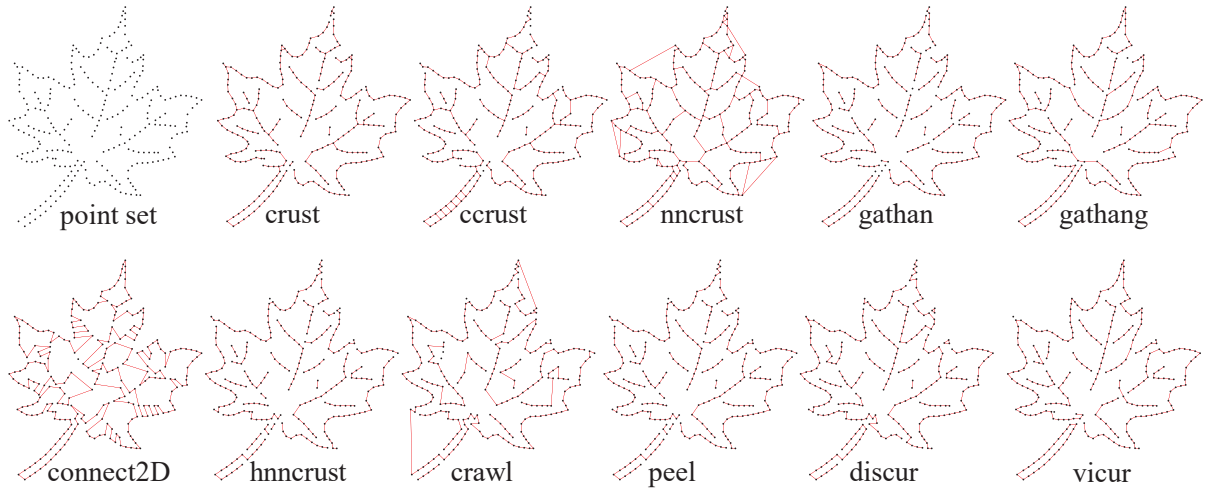

Figure 15: *A qualitative comparison of different algorithms on the leaf input that contains sharp-corners, non-manifold edges, multiple and open curves (*run-qualitative.sh*).*

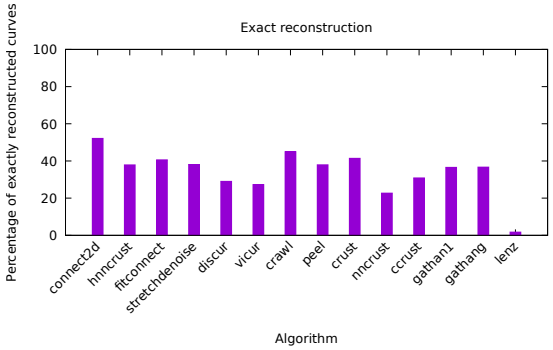

Figure 16: *Percentage of exactly reconstructed curves from a set of 2183 noise-free point sets predominantly generated from images, for 14 algorithms (*run-manifold.sh*).*

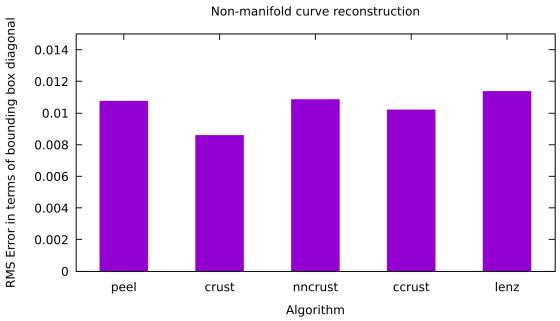

Figure 17: *RMS error of reconstructed non-manifold curves compared with ground truth. We tested these algorithms on 26 inputs which consist of 16 curves collected from various curve reconstruction papers and 10 synthetically created curves (*run-non-manifold.sh*).*

correctly, followed by GATHAN1 with a success rate of 44.68%. The remaining algorithms succeed only for few cases, i.e., [0, 23.40]%.

6.2.4. Open Curves

Figure 19 shows a bar diagram with the percentage of correctly reconstructed open curves per algorithm along the y-axis. Input to the algorithms wass a set of 23 open curve point sets manually selected from various reconstruction papers. We excluded CONNECT2D and STRETCHDENOISE because both these algorithms were not designed for open curves. The VICUR algorithm reconstructs most (52.2%) of the open curves. Other algorithms such as HNNCRUST, CRUST, PEEL, and DISCUR also perform reasonably well on open curves.

6.2.5. Multiple Curves

We considered two types of inputs in this experiment: multiply connected curves (curves with holes) and disconnected curves. We evaluated 11 algorithms on 54 point sets. We omitted CONNECT2D, STRETCHDENOISE, and LENZ because these algorithms were not designed to handle multiple curves. Figure 20 shows the bar diagram displaying the percentage of curves reconstructed per algorithm along the y-axis. In our experiment, PEEL and HNN-CRUST handled multiple curves comparatively well.

6.3. Guarantees as Sampling Conditions

For some algorithms, reconstruction is guaranteed if a certain sampling condition is fulfilled, this specifies, e.g., with which maximum distance the points are allowed to be sampled on the original curve, often depending on some other criteria, but other conditions also exist. Older algorithms like α-shapes based Ball-pivoting [BB97] require a globally uniform maximum distance for samples. Most of the algorithms compared here in Table 4 rely on the ε-sampling condition, which depends on the local size of features. Among these, HNN-CRUST performs best with $\varepsilon < 0.47$. ε-sampling can be relaxed if other conditions are added, such as an angle condition

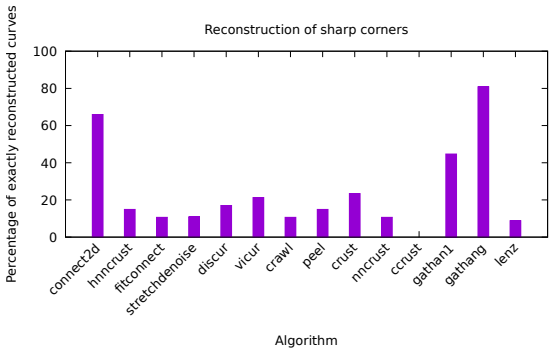

Figure 18: *Percentage of exactly reconstructed curves having sharp corners by various algorithms. We tested the algorithms on 47 inputs which consist of 21 curves collected from various curve reconstruction papers and 26 synthetically created curves (*run-sharp-corners.sh*).*

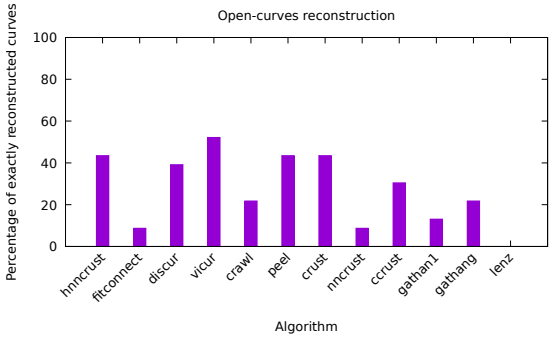

Figure 19: *Percentage of exact reconstruction of open curves by different algorithms. We tested the algorithms on 23 different open curves collected from various curve reconstruction papers (*run-open-curves.sh*).*

(GATHANG) or a maximum factor between adjacent edge lengths (CONNECT2D). Ohrhallinger et al. [OMW16] propose a new kind of sampling condition, ρ-sampling, which applies to curve intervals instead of curve points, and therefore permits a sparser sampling and thus fewer required points than the ε-sampling they show it correlates with. Funke et al. [FR01] design a sampling condition explicitly for sharp corners (algorithm not compared here), but it is very complex and requires 8 parameters. DISCUR uses a complex vision function as sampling condition. We observed that often curves can be successfully reconstructed from much sparser sampling than proven for the algorithms across the field. This indicates that sampling conditions could still be improved.

6.4. Summary

Figures 21 and 22 show manually chosen examples of successful and failed reconstructions for all the 15 algorithms reported in the respective papers. Failure cases represent worst-case point sets for various aspects such as noise-free, sparsely-sampled, noisy, outliers, noisy+outliers, non-manifold, sharp corners, etc. We report the

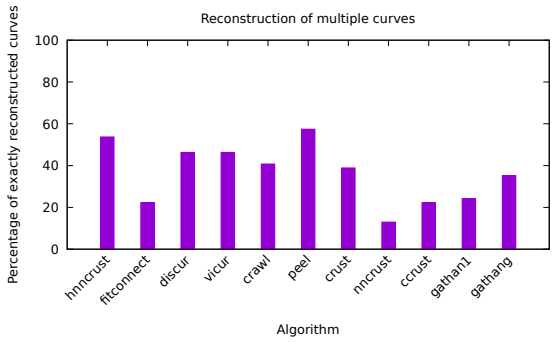

Figure 20: *Percentage of exactly reconstructed curves (multiply connected as well as disconnected curves) by various algorithms. We tested the algorithms on 52 different point sets collected from various curve reconstruction papers, and 2 point sets sampled from image silhouette boundaries (*run-multiple-curves.sh*).*

Algorithm	Sampling condition
CRUST [ABE98]	$\varepsilon < 0.252$
NN-CRUST [DK99]	$\varepsilon < \frac{1}{3}$
CCRUST [DMR99]	$\varepsilon < 0.0625$
GATHAN [DW01]	none
GATHANG [DW02]	$\varepsilon < 0.5, \alpha > 150°$
LENZ [Len06]	$\varepsilon < 0.48$ (no proof)
CONNECT2D [OM13]	$\varepsilon < \frac{1}{2}, u < 1.609$
CRAWL [PM16]	none
HNN-CRUST [OMW16]	$\varepsilon < 0.47 \equiv \rho < 0.9$
FITCONNECT [OW18a]	as HNN-CRUST
STRETCHDENOISE [OW18b]	as HNN-CRUST
PEEL [PMM18]	none
OPT.TRANS. [DGCSAD11]	none
DISCUR [ZNYL08]	vision function
VICUR [NZ08]	none

Table 4: *Sampling conditions of algorithms w.r.t. reconstruction of a manifold curve, where applicable.* ε *refers to* ε*-sampling [ABE98],* α *to an opening angle between adjacent edges, u to local uniformity as factor of adjacent edge lengths.*

best two algorithms for handling specific types of input and curve characteristics in Table 5.

7. Conclusion

After analyzing the properties of 36 algorithms and comparing up to 15 of them, we showed that there are quite diverse approaches for curve reconstruction. This stems from the various challenges in the field, ranging from connecting uniformly distributed feature points over noisy silhouettes from sensors, non-smooth curves to non-uniform sampling and sketches which do not represent an outline. In order to choose a suitable algorithm for a specific application, the user can first narrow the field by availability (as open source) and category, which roughly corresponds to the above applications, then select among those based on importance regarding the criteria compared in our evaluation. In our survey, we also describe the

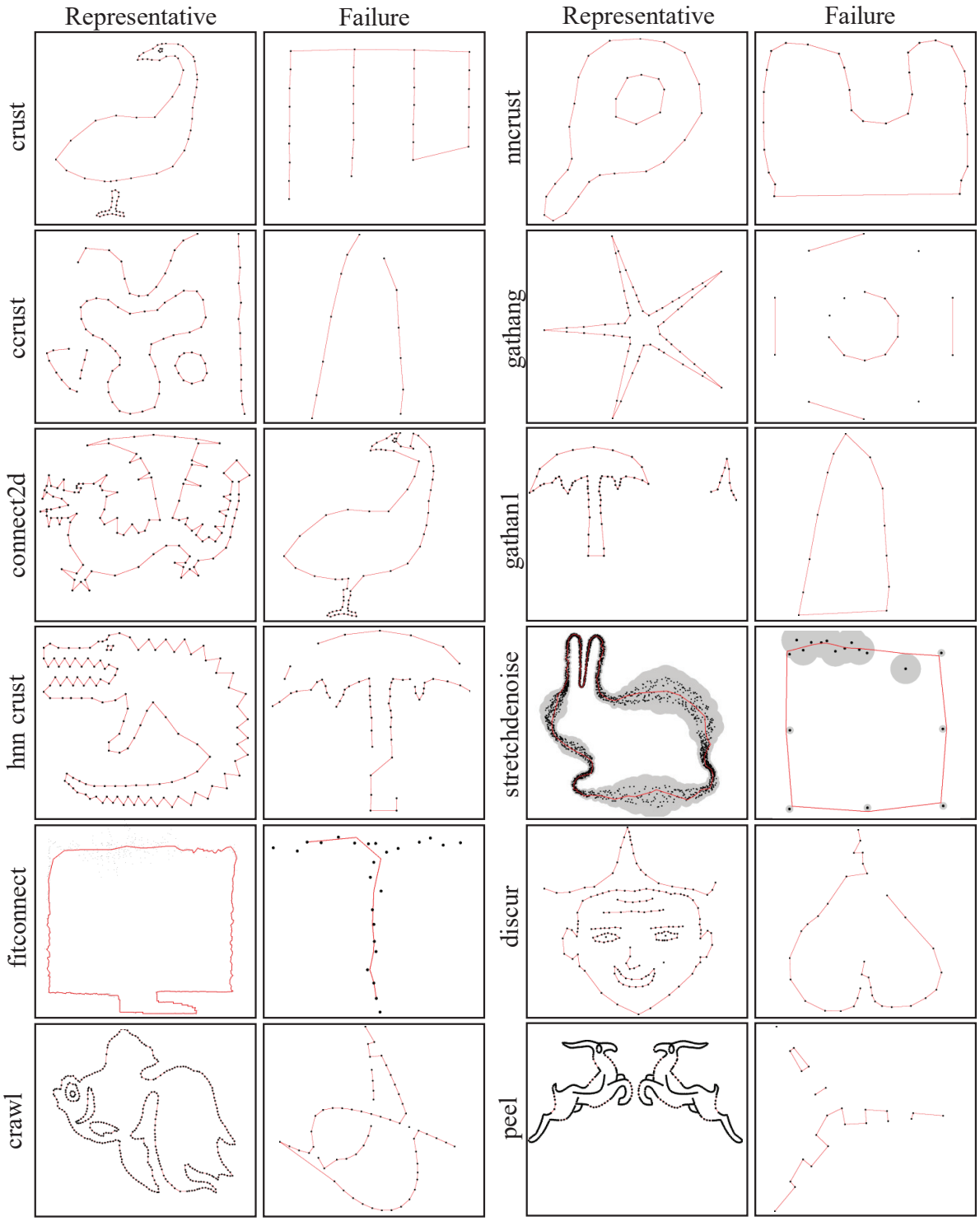

Figure 21: *Representative and failed results of various curve reconstruction algorithms (part 1).*

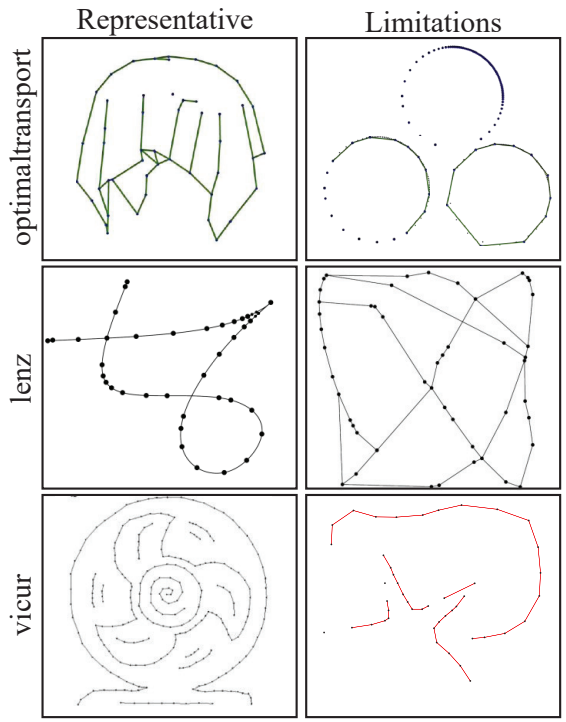

Figure 22: *Representative and failed results (part 2) of* OPTIMALTRANSPORT *(row 1),* LENZ *(row 2) and* VICUR *(row 3). Image courtesy: row 1 [DGCSAD11], row 2 [Len06] and row 3 (left) [NZ08]. Note that all three algorithms are sensitive to lower sampling density.*

Curve/Input feature	Best two algorithms in order
Uniform Noise	DISCUR, VICUR
Non-uniform Noise	STRETCHDENOISE, CONNECT2D
Outliers	HNN-CRUST, CRUST
Non-uniform sampling	HNN-CRUST, PEEL
Runtime	NN-CRUST, GATHAN1
Manifold curves	CONNECT2D, CRAWL
Non-manifold curves	CRUST, LENZ
Sharp features	GATHANG, CONNECT2D
Open curves	VICUR, HNN-CRUST
Multiple curves	PEEL, HNN-CRUST

Table 5: *Summary of the experimental study. We list the best and second-best algorithms in each category of input or curve characteristic. Note that a few algorithms were not included in these experiments due to their unavailability as open source, and hence, the proposals that we make here are not comprehensive. The two fastest algorithms were reported based on the manifold curve reconstruction experiment on a test set consisting of 2183 inputs.*

evolution of the algorithms, so some are simply outperformed by later ones. In terms of theoretical guarantees, the potential of ε-sampling only being based on local feature size seems to be maxed out.

7.1. Future Directions

In this mature field, we still consider some directions to be worthwhile for future work, mostly building onto this fundamental research surveyed here.

- Improving and simplifying sampling conditions, especially for non-smooth and self-intersecting curves
- Reconstructing curves from hand-drawn sketches with varying stroke thickness and intensity
- Deep learning on curves, as for surface reconstruction
- Reconstruct smooth curves instead of (open) polygons
- Reconstruction of surfaces from networks of 3D curves

Acknowledgements

This work has been partially funded by the Austrian Science Fund (FWF) projects no. P24600-N23 and P32418-N31, by the Wiener Wissenschafts-, Forschungs- und Technologiefonds (WWTF) project ICT19-009 and by the NSERC Discovery grant #2019-05092.

References

[107] The 1070-shape database. http://vision.lems.brown.edu/sites/default/files/1070db.tar.gz. [Online; accessed 19-October-2019]. 11

[AAA*09] AICHHOLZER O., AIGNER W., AURENHAMMER F., HACKL T., JÜTTLER B., RABL M.: Medial axis computation for planar free-form shapes. *Computer-Aided Design 41*, 5 (2009), 339 – 349. Voronoi Diagrams and their Applications. 12

[ABCC] APPLEGATE D., BIXBY R., CHVATAL V., COOK W.: Concorde tsp solver. http://www.math.uwaterloo.ca/tsp/concorde.html. [Online; accessed 12-September-2018]. 3, 9

[ABCo*03] ALEXA M., BEHR J., COHEN-OR D., FLEISHMAN S., LEVIN D., SILVA C. T.: Computing and rendering point set surfaces. *IEEE Transactions on Visualization and Computer Graphics 9* (2003), 3–15. 10

[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, 5, 7, 9, 11, 13, 15, 17

[ABE09] ATTALI D., BOISSONNAT J.-D., EDELSBRUNNER H.: Stability and computation of medial axes-a state-of-the-art report. In *Mathematical foundations of scientific visualization, computer graphics, and massive data exploration*. Springer, 2009, pp. 109–125. 12

[ABK98] AMENTA N., BERN M., KAMVYSSELIS M.: A new voronoi-based surface reconstruction algorithm. In *Proceedings of the 25th annual conference on Computer graphics and interactive techniques* (1998), ACM, pp. 415–421. 7

[ACDL00] AMENTA N., CHOI S., DEY T. K., LEEKHA N.: A simple algorithm for homeomorphic surface reconstruction. In *Proceedings of the sixteenth annual symposium on Computational geometry* (2000), ACM, pp. 213–222. 7

[ACSd*18] ALLIEZ P., COHEN-STEINER D., DEGOES F., JAMIN C., VIGAN I.: Optimal transportation reconstruction, 2018. 11

[ACSTD07] ALLIEZ P., COHEN-STEINER D., TONG Y., DESBRUN M.: Voronoi-based variational reconstruction of unoriented point sets. In *Symposium on Geometry processing* (2007), vol. 7, pp. 39–48. 10

[AD15] AKIMALIEV M., DEMIRCI M. F.: Improving skeletal shape abstraction using multiple optimal solutions. *Pattern Recognition 48*, 11 (2015), 3504 – 3515. 11

[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), Soc. for Industr. and Appl. Math., pp. 686–695. 3, 9

[AMS00] ALTHAUS E., MEHLHORN K., SCHIRRA S.: Experiments on curve reconstruction. In *Proc. 2nd Workshop Algorithm Eng. Exper* (2000), pp. 103–114. 2, 9

[Aro98] ARORA S.: Polynomial time approximation schemes for euclidean traveling salesman and other geometric problems. *Journal of the ACM (JACM) 45*, 5 (1998), 753–782. 3, 9

[ARZ05] AHLVERS U., RAJAGOPALAN R., ZÖLZER U.: Model-free face detection and head tracking with morphological hole mapping. In *2005 13th European Signal Processing Conference* (2005), IEEE, pp. 1–4. 1

[Att97] ATTALI D.: r-regular shape reconstruction from unorganized points. In *Symp. on Computational Geometry* (1997), pp. 248–253. 3, 7

[BB97] BERNARDINI F., BAJAJ C. L.: Sampling and reconstructing manifolds using alpha-shapes. *Proc. 9th Canad. Conf. Comput. Geom.* (1997). 3, 6, 16

[BLN*13] BERGER M., LEVINE J. A., NONATO L. G., TAUBIN G., SILVA C. T.: A benchmark for surface reconstruction. *ACM Transactions on Graphics (TOG) 32*, 2 (2013), 20. 12

[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. 5

[BMR*99] BERNARDINI F., MITTLEMAN J., RUSHMEIER H., SILVA C., TAUBIN G.: The ball-pivoting algorithm for surface reconstruction. *IEEE transactions on visualization and computer graphics 5*, 4 (1999), 349–359. 6, 7

[Boi84a] BOISSONAT J.: Representing 2d and 3d shapes with the delaunay triangulation. In *InSeventh International Conference on Pattern Recognition (ICPR'84)* (1984). 6, 7

[Boi84b] BOISSONNAT J.-D.: Geometric structures for three-dimensional shape representation. *ACM Transactions on Graphics (TOG) 3*, 4 (1984), 266–286. 9

[Boo79] BOOKSTEIN F. L.: Fitting conic sections to scattered data. *Computer Graphics and Image Processing 9*, 1 (1979), 56–71. 10

[CBC*01] CARR J. C., BEATSON R. K., CHERRIE J. B., MITCHELL T. J., FRIGHT W. R., MCCALLUM B. C., EVANS T. R.: Reconstruction and representation of 3d objects with radial basis functions. In *Proceedings of the 28th Annual Conference on Computer Graphics and Interactive Techniques* (New York, NY, USA, 2001), SIGGRAPH '01, ACM, pp. 67–76. 10

[CFG*05] CHENG S.-W., FUNKE S., GOLIN M., KUMAR P., POON S.-H., RAMOS E.: Curve reconstruction from noisy samples. *Computational Geometry 31*, 1-2 (2005), 63–100. 3, 8

[dALJ*15] DE ARAÚJO B. R., LOPES D. S., JEPP P., JORGE J. A., WYVILL B.: A survey on implicit surface polygonization. *ACM Comput. Surv. 47*, 4 (May 2015). 10

[Dey06] DEY T. K.: *Curve and surface reconstruction: algorithms with mathematical analysis*, vol. 23. Cambridge University Press, 2006. 2

[DGCSAD11] DE GOES F., COHEN-STEINER D., ALLIEZ P., DESBRUN M.: An optimal transport approach to robust reconstruction and simplification of 2d shapes. In *Computer Graphics Forum* (2011), vol. 30, Wiley Online Library, pp. 1593–1602. 3, 9, 11, 13, 17, 19

[DK99] DEY T. K., KUMAR P.: A simple provable algorithm for curve reconstruction. In *Proc. 10th ACM-SIAM SODA '99* (1999), pp. 893–894. 3, 7, 9, 11, 15, 17

[DKWG08] DUCKHAM M., KULIK L., WORBOYS M., GALTON A.: Efficient generation of simple polygons for characterizing the shape of a set of points in the plane. *Pattern Recognition 41*, 10 (2008), 3224 – 3236. 2

[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. 3, 7, 11, 15, 17

[DT14] DUARTE P., TORRES M. J.: Smoothness of boundaries of regular sets. *Journal of mathematical imaging and vision* (2014), 1–8. 5

[DW01] DEY T. K., WENGER R.: Reconstructing curves with sharp corners. *Computational Geometry 19*, 2-3 (2001), 89 – 99. 3, 8, 11, 15, 17

[DW02] DEY T. K., WENGER R.: Fast reconstruction of curves with sharp corners. *Int. J. Comp. Geom. Appl. 12*, 5 (2002), 353 – 400. 3, 9, 11, 15, 17

[Ede98] EDELSBRUNNER H.: Shape reconstruction with delaunay complex. In *LATIN'98: Theoretical Informatics* (Berlin, Heidelberg, 1998), Lucchesi C. L., Moura A. V., (Eds.), Springer Berlin Heidelberg, pp. 119–132. 2

[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, 3, 6

[EM94] EDELSBRUNNER H., MÜCKE E. P.: Three-dimensional alpha shapes. *ACM Transactions on Graphics (TOG) 13*, 1 (1994), 43–72. 6

[FCOAS03] FLEISHMAN S., COHEN-OR D., ALEXA M., SILVA C. T.: Progressive point set surfaces. *ACM Trans. Graph. 22*, 4 (Oct. 2003), 997–1011. 10

[FCOS05] FLEISHMAN S., COHEN-OR D., SILVA C. T.: Robust moving least-squares fitting with sharp features. *ACM Trans. Graph. 24*, 3 (July 2005), 544–552. 10

[Fed59] FEDERER H.: Curvature measures. *Transactions of the American Mathematical Society 93*, 3 (1959), pp. 418–491. 5

[FMG94] FIGUEIREDO L. H. D., MIRANDAS GOMES J. D.: Computational morphology of curves. *Vis. Comp. 11*, 2 (1994), 105–112. 3, 6, 7

[FMZB91] FORSYTH D., MUNDY J. L., ZISSERMAN A., BROWN C. M.: Projectively invariant representations using implicit algebraic curves. *Image and Vision Computing 9*, 2 (1991), 130–136. 10

[FR01] FUNKE S., RAMOS E. A.: Reconstructing a collection of curves with corners and endpoints. In *Proceedings of the twelfth annual ACM-SIAM symposium on Discrete algorithms* (Philadelphia, PA, USA, 2001), SODA '01, Society for Industrial and Applied Math., pp. 344–353. 3, 9, 17

[GDJ*11] GHEIBI A., DAVOODI M., JAVAD A., PANAHI F., AGHDAM M., ASGARIPOUR M., MOHADES A.: Polygonal shape reconstruction in the plane. *Computer Vision, IET 5*, 2 (March 2011), 97 –106. 2

[Gen90] GENOUD T.: Etude du caractère hamiltonien de delaunays aléatoires. Travail de semestre, 1990. 6

[GG07] GUENNEBAUD G., GROSS M.: Algebraic point set surfaces. *ACM Transactions on Graphics (TOG) 26*, 3 (2007), 23. 10

[Gie99] GIESEN J.: Curve reconstruction in arbitrary dimension and the traveling salesman problem. In *Proc. 8th DCGI '99* (1999), pp. 164–176. 3, 9

[Gol99] GOLD C.: Crust and anti-crust: a one-step boundary and skeleton extraction algorithm. In *Proc. of the 15th ann. Symp. on Computational geometry* (New York, NY, USA, 1999), SCG '99, ACM, pp. 189–196. 3, 7

[HDD*92] HOPPE H., DEROSE T., DUCHAMP T., MCDONALD J., STUETZLE W.: Surface reconstruction from unorganized points. *SIGGRAPH Comput. Graph. 26*, 2 (July 1992), 71–78. 10

[Hiy09] HIYOSHI H.: Optimization-based approach for curve and surface reconstruction. *Comput. Aided Des. 41* (May 2009), 366–374. 3, 8, 11

[HS09] HOOS H. H., STÜTZLE T.: *On the empirical scaling of run-time for finding optimal solutions to the traveling salesman problem*. Tech. rep., Technical Report 17, University of British Columbia, Department of Computer Science, 2009. 9

[IG16] IGLESIAS A., GÁLVEZ A.: Cuckoo search with lévy flights for reconstruction of outline curves of computer fonts with rational bézier curves. In *2016 IEEE Congress on Evolutionary Computation (CEC)* (2016), pp. 2247–2254. 8

[IGA18] IGLESIAS A., GALVEZ A., AVILA A.: Immunological approach for full nurbs reconstruction of outline curves from noisy data points in medical imaging. *IEEE/ACM Trans. Comput. Biol. Bioinformatics 15*, 6 (Nov. 2018), 1929–1942. 8

[Jar77] JARVIS R.: Computing the shape hull of points in the plane. In *Proceedings of the IEEE Computing Society Conference on Pattern Recognition and Image Processing* (1977), New York, pp. 231–241. 6

[JT92] JAROMCZYK J., TOUSSAINT G.: Relative neighborhood graphs and their relatives. *Proceedings of the IEEE 80*, 9 (sep 1992), 1502 –1517. 6

[Kaz05] KAZHDAN M.: Reconstruction of solid models from oriented point sets. In *Proceedings of the Third Eurographics Symposium on Geometry Processing* (Goslar, DEU, 2005), SGP '05, Eurographics Association, p. 73–es. 10

[KBH06] KAZHDAN M., BOLITHO M., HOPPE H.: Poisson surface reconstruction. In *Proceedings of the Fourth Eurographics Symposium on Geometry Processing* (Goslar, DEU, 2006), SGP '06, Eurographics Association, p. 61–70. 10

[Kol08] KOLLURI R.: Provably good moving least squares. *ACM Trans. Algorithms 4*, 2 (May 2008), 18:1–18:25. 10

[KR85] KIRKPATRICK D. G., RADKE J. D.: A framework for computational morphology. *Computational Geometry* (1985), 217–248. 3, 6

[KR14] KHANNA K., RAJPAL N.: Survey of curve and surface reconstruction algorithms from a set of unorganized points. In *Proceedings of the Third International Conference on Soft Computing for Problem Solving* (2014), Springer, pp. 451–458. 2

[KTB07] KATZ S., TAL A., BASRI R.: Direct visibility of point sets. In *ACM SIGGRAPH 2007 papers*. 2007, pp. 24–es. 8

[LC87] LORENSEN W. E., CLINE H. E.: Marching cubes: A high resolution 3d surface construction algorithm. *SIGGRAPH Comput. Graph. 21*, 4 (Aug. 1987), 163–169. 10

[Lee00a] LEE I.-K.: Curve reconstruction from unorganized points. *Computer aided geometric design 17*, 2 (2000), 161–177. 3, 8, 11

[Lee00b] LEE I.-K.: Curve reconstruction from unorganized points. *Computer Aided Geometric Design 17*, 2 (2000), 161 – 177. 10

[Len06] LENZ T.: How to sample and reconstruct curves with unusual features. In *Proceedings of the 22nd European Workshop on Computational Geometry (EWCG)* (Delphi, Greece, March 2006). 3, 8, 11, 15, 17, 19

[Lev98] LEVIN D.: The approximation power of moving least-squares. *Mathematics of Computation of the American Mathematical Society 67*, 224 (1998), 1517–1531. 8

[Lev04] LEVIN D.: *Mesh-Independent Surface Interpolation*. Springer Berlin Heidelberg, Berlin, Heidelberg, 2004, pp. 37–49. 10

[Mat16] MATHER G.: *Foundations of Sensation and Perception*. Taylor & Francis, 2016. 9

[MKPM17] METHIRUMANGALATH S., KANNAN S. S., PARAKKAT A. D., MUTHUGANAPATHY R.: Hole detection in a planar point set: An empty disk approach. *Computers & Graphics 66* (2017), 124 – 134. Shape Modeling International 2017. 3

[mpe02] Mpeg-7 core experiment ce-shape-1 test set. https://cis.temple.edu/~latecki/TestData/mpeg7shapeB.tar.gz, 2002. [Online; accessed 19-October-2019]. 11

[MPM15] METHIRUMANGALATH S., PARAKKAT A. D., MUTHUGANAPATHY R.: A unified approach towards reconstruction of a planar point set. *Computers & Graphics 51* (2015), 90 – 97. International Conference Shape Modeling International. 2

[MPS08] MANSON J., PETROVA G., SCHAEFER S.: Streaming surface reconstruction using wavelets. *Computer Graphics Forum 27*, 5 (2008), 1411–1420. 10

[MTSM10] MEHRA R., TRIPATHI P., SHEFFER A., MITRA N. J.: Visibility of noisy point cloud data. *Computers & Graphics 34*, 3 (2010), 219–230. 3, 8, 11, 14

[NSW08] NIYOGI P., SMALE S., WEINBERGER S.: Finding the homology of submanifolds with high confidence from random samples. *Discrete & Computational Geometry 39*, 1-3 (2008), 419–441. 7

[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 International Conference on Flexible Automation and Intelligent Manufacturing. 3, 9, 11, 15, 17, 19

[OBA*03] OHTAKE Y., BELYAEV A., ALEXA M., TURK G., SEIDEL H.-P.: Multi-level partition of unity implicits. *ACM Trans. Graph. 22*, 3 (July 2003), 463–470. 10

[OBS06] OHTAKE Y., BELYAEV A., SEIDEL H.-P.: Sparse surface reconstruction with adaptive partition of unity and radial basis functions. *Graphical Models 68*, 1 (2006), 15 – 24. Special Issue on SMI 2004. 10

[OBW87] O'ROURKE J., BOOTH H., WASHINGTON R.: Connect-the-dots: a new heuristic. *Computer Vision, Graphics, and Image Processing 39*, 2 (1987), 258–266. 7

[Ohr13] OHRHALLINGER S.: An efficient algorithm for determining an aesthetic shape connecting unorganized 2d points. https://sourceforge.net/projects/connect2dlib/, 2013. [Online; accessed 12-September-2018]. 2, 11

[Ohr16] OHRHALLINGER S.: Half-nearest neighbor crust. https://github.com/stefango74/hnn-crust-sgp16, 2016. [Online; accessed 12-September-2018]. 11

[Ohr18a] OHRHALLINGER S.: Fitconnect. https://github.com/stefango74/fitconnect, 2018. [Online; accessed 12-September-2018]. 11

[Ohr18b] OHRHALLINGER S.: Stretchdenoise. https://github.com/stefango74/stretchdenoise, 2018. [Online; accessed 12-September-2018]. 11

[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. 3, 7, 10

[OM13] OHRHALLINGER S., MUDUR S.: An efficient algorithm for determining an aesthetic shape connecting unorganized 2d points. In *Comp. Graph. Forum* (2013), vol. 32, Wiley Online Library, pp. 72–88. 3, 7, 10, 11, 15, 17

[OMW16] OHRHALLINGER S., MITCHELL S. A., WIMMER M.: Curve reconstruction with many fewer samples. In *Computer Graphics Forum* (2016), vol. 35, Wiley Online Library, pp. 167–176. 2, 3, 5, 8, 11, 15, 17

[O'R81] O'ROURKE J.: Polyhedra of minimal area as 3d object models. In *IJCAI* (1981), Citeseer, pp. 664–666. 9

[OW18a] OHRHALLINGER S., WIMMER M.: Fitconnect: Connecting noisy 2d samples by fitted neighbourhoods. In *Computer Graphics Forum* (2018), Wiley Online Library. 2, 3, 4, 8, 11, 15, 17

[OW18b] OHRHALLINGER S., WIMMER M.: Stretchdenoise: Parametric curve reconstruction with guarantees by separating connectivity from residual uncertainty of samples. In *Pacific Graphics* (2018), Wiley Online Library. 3, 4, 8, 11, 15, 17

[Par16] PARAKKAT A. D.: Crawl through neighbors. https://github.com/reproducibilitystamp/Crawl_through_Neighbors, 2016. [Online; accessed 12-September-2018]. 11

[Par18] PARAKKAT A. D.: Peeling the longest. https://github.com/amaldevp/Peeling_the_Longest, 2018. [Online; accessed 12-September-2018]. 11

[PM15a] PEETHAMBARAN J., MUTHUGANAPATHY R.: A non-parametric approach to shape reconstruction from planar point sets through delaunay filtering. *Computer-Aided Design 62* (2015), 164 – 175. 2

[PM15b] PEETHAMBARAN J., MUTHUGANAPATHY R.: Reconstruction of water-tight surfaces through delaunay sculpting. *Computer-Aided Design 58* (2015), 62 – 72. Solid and Physical Modeling 2014. 3, 5, 7

[PM16] PARAKKAT A. D., MUTHUGANAPATHY R.: Crawl through neighbors: A simple curve reconstruction algorithm. In *Computer Graphics Forum* (2016), vol. 35, Wiley Onl. Library, pp. 177–186. 2, 3, 7, 11, 15, 17

[PMM18] PARAKKAT A. D., METHIRUMANGALATH S., MUTHUGANAPATHY R.: Peeling the longest: A simple generalized curve reconstruction algorithm. *Computers & Graphics 74* (2018), 191 – 201. 3, 9, 11, 15, 17

[PPT*19] PEETHAMBARAN J., PARAKKAT A., TAGLIASACCHI A., WANG R., MUTHUGANAPATHY R.: Incremental labelling of voronoi vertices for shape reconstruction. *Computer Graphics Forum 0*, 0 (2019). 2, 3, 5, 7

[Pra87] PRATT V.: Direct least-squares fitting of algebraic surfaces. *SIGGRAPH Comput. Graph. 21*, 4 (Aug. 1987), 145–152. 10

[Roh20] ROHATGI A.: Webplotdigitizer: Version 4.3, 2020. 11

[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. 5, 7

[Rup14] RUPNIEWSKI M. W.: Curve reconstruction from noisy and unordered samples. In *3rd International Conference on Pattern Recognition Applications and Methods* (2014), SciTePress. 3, 8

[SJP10] SONG X., JÜTTLER B., POTEAUX A.: Hierarchical spline approximation of the signed distance function. In *2010 Shape Modeling International Conference* (2010), pp. 241–245. 10

[ST09] STELLDINGER P., TCHERNIAVSKI L.: Provably correct reconstruction of surfaces from sparse noisy samples. *Pattern Recognition 42*, 8 (2009), 1650–1659. 7

[Ste08] STELLDINGER P.: Topologically correct surface reconstruction using alpha shapes and relations to ball-pivoting. In *Pattern Recognition, 2008. ICPR 2008. 19th Int'l Conference on* (2008), IEEE, pp. 1–4. 7

[Tau91] TAUBIN G.: Estimation of planar curves, surfaces, and nonplanar space curves defined by implicit equations with applications to edge and range image segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence 13*, 11 (1991), 1115–1138. 10

[Tau93] TAUBIN G.: An improved algorithm for algebraic curve and surface fitting. In *1993 (4th) International Conference on Computer Vision* (1993), pp. 658–665. 10

[TO02] TURK G., O'BRIEN J. F.: Modelling with implicit surfaces that interpolate. *ACM Trans. Graph. 21*, 4 (Oct. 2002), 855–873. 10

[Tou88] TOUSSAINT G.: A graph-theoretical primal sketch. *Computational Morphology* (1988), 229–260. 6

[TPM20] THAYYIL S. B., PARAKKAT A. D., MUTHUGANAPATHY R.: An input-independent single pass algorithm for reconstruction from dot patterns and boundary samples. *Computer Aided Geometric Design 80* (2020), 101879. 2

[TPM21] THAYYIL S. B., PEETHAMBARAN J., MUTHUGANAPATHY R.: A sampling type discernment approach towards reconstruction of a point set in r2. *Computer Aided Geometric Design 84* (2021), 101953. 2

[Vel92] VELTKAMP R. C.: The γ-neighborhood graph. *Computational Geometry 1*, 4 (1992), 227–246. 6

[Vel93] VELTKAMP R. C.: 3d computational morphology. In *Computer Graphics Forum* (1993), vol. 12, Wiley Online Library, pp. 115–127. 3, 6

[WYZ*14] WANG J., YU Z., ZHANG W., WEI M., TAN C., DAI N., ZHANG X.: Robust reconstruction of 2d curves from scattered noisy point data. *Computer-Aided Design 50* (2014), 27–40. 3, 4, 8, 11

[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. 3, 9, 11, 15, 17