An Optimal Transport Approach to Robust Reconstruction and Simplification of 2D Shapes

Thesis by

Fernando Ferrari de Goes

In Partial Fulfillment of the Requirements

for the Degree of

Master of Science

California Institute of Technology

Pasadena, California

2011

(Submitted May 22, 2011)

Contents

Acknowledgements

In one of my first days in Rio, my former advisor, Prof. Luiz Velho, was talking with some students about meshing. At that time, I was an undergrad doing a summer internship in the VISGRAF lab. In his explanation, Luiz was referring to a sequence of articles written by three authors. I took note of their names: Mathieu Desbrun, Pierre Alliez and David Cohen-Steiner. As unpredictable as life can be, this is my second work with them. Mathieu, Pierre and David, thanks a lot for all the advices, support, encouragement and enthusiasm. I have had a great time!!!

I also want to thank Patrick Mullen, Pooran Memari and Lance Williams for their comments, and Doug DeCarlo, Xiaofeng Mi, and Ravish Mehra for data.

Abstract

We present a robust 2D shape reconstruction and simplification algorithm which takes as input a defect-laden point set with noise and outliers. We introduce an optimal-transport driven approach where the input point set, considered as a sum of Dirac measures, is approximated by a simplicial complex considered as a sum of uniform measures on 0- and 1-simplices. A fine-to-coarse scheme is devised to construct the resulting simplicial complex through greedy decimation of a Delaunay triangulation of the input point set. Our method performs well on a variety of examples ranging from line drawings to grayscale images, with or without noise, features, and boundaries.

Chapter 1

Introduction

Shape reconstruction from unorganized point sets is a fundamental problem in geometry processing: despite significant advances, its inherent ill-posedness and the increased heterogeneity of geometric datasets contribute to make current approaches still far from satisfactory. Even the 2D instance of this problem, i.e., the reconstruction of shapes in the plane, remains a challenge in various application areas including computer vision and image processing. Two-dimensional point sets are often acquired from sensors or extracted from images, and are thus frequently hampered with aliasing, noise, and outliers. In addition, image-based point sets often exhibit a rich variety of features such as corners, intersections, bifurcations, and boundaries. This combination of noise, outliers, and presence of boundaries and features render most well-known strategies (including Poisson, Delaunay, or MLS-based approaches) deficient.

Shape reconstruction is also intimately linked to shape simplification. While a few authors (in particular in Computational Geometry) have restricted the issue of reconstruction to finding a connectivity of *all* the input points, the presence of noise and the sheer size of most datasets require the final reconstructed shape to be more concise than the input. Reconstruction and simplification are, however, often performed sequentially rather than in concert.

In this thesis, we jointly address reconstruction and simplification of 2D shapes through a unified framework rooted in optimal transport of measures. Specific benefits include (i) robustness to large amounts of noise and outliers; (ii) preservation of sharp features; (iii) preservation of boundaries; and (iv) guarantee that the output is a (possibly non-manifold) embedded simplicial complex.

1.1 Previous Work

To motivate our approach and stress how it fills theoretical and practical needs, we first review previous work in both reconstruction and simplification of 2D point sets.

1.1.1 Reconstruction.

For noise free datasets, existing reconstruction methods vary mostly based on sampling assumptions. Uniformly sampled datasets can be dealt with using image thinning [31], alpha shapes [17] or r-regular shapes [4]; for non-uniform sampling, most provably correct methods rely on Delaunay filtering [3], with improvements on computational efficiency [21], sampling bounds [13, 14], and handling of corners and open curves [19, 15].

Noisy datasets have been tackled by a variety of methods over the last ten years [26, 9, 33, 30], with recent successes including the extraction of self-intersecting curves [38]. Most of these approaches remove noise through clustering, thinning, or averaging before performing reconstruction, which often leads to a significant blunting of features. Robustness to outliers has also been—to a lesser extent—investigated, with approaches ranging from data clustering [39] and robust statistics [18], to k-order alpha shapes [25], spectral methods [24] and ℓ_1-minimization [5]—but often at the cost of a loss of sharp features and/or a significant increase in computational complexity.

1.1.2 Simplification.

Polygonal curve simplification has received attention from many fields, including cartography, computer graphics, computer vision and other medical and scientific imaging applications. A common approach to simplification is to proceed in a fine-to-coarse fashion through decimation [1, 20], whereas coarse-to-fine methods—such as the Douglas-Peucker algorithm [22]—proceed through refinement instead. When topological guarantees are called for, a fine-to-coarse strategy is often preferred; for example, Dyken et al. [16] used intersection tests at each decimation step to preserve nesting of cartographic contours. Topological guarantees are indeed more difficult to obtain in the coarse-to-fine strategy, especially when sharp angles and singularities are present.

Simplification methods can also be classified according to the metric they use to measure the accuracy of the simplified curves. Most methods are based on the Hausdorff distance or the Fréchet distance (see [1] and the references therein). However, these metrics do not handle noise well, and significant processing is required to filter spurious data.

A distinct, but related endeavor is to concisely convey semantic aspects of a geometric dataset. This problem is usually referred to as *abstraction* [32], and is of interest to applications in shape recognition and processing. In [10], we investigated the problem of abstracting 3D surfaces as a network of curves and presented a principled framework that combines structural segmentation and shape approximation.

While specific data idiosyncrasies such as noise, outliers, features, and boundaries have been successfully dealt with individually, little or none of the previous work can handle them all concurrently.

1.2 Contributions

We depart from previous work by leveraging the versatile framework of optimal transport: we view shape reconstruction and simplification as a transport problem between measures (i.e., mass distributions), where the input points are considered as Dirac measures and the reconstructed simplicial complex is seen as the support of a piecewise uniform measure. The use of optimal transport brings forth several benefits, including a unified treatment of noise, outliers, boundaries, and sharp features. Our reconstruction algorithm derives a simplicial complex by greedily minimizing the symmetric Wasserstein transport metric between the Dirac masses that the input points represent and a piecewise uniform measure on the 0- and 1-simplices of the reconstructed complex. A fine-to-coarse strategy is proposed for efficiency, starting from the 2D Delaunay triangulation of the input point set and proceeding through repeated edge collapses and vertex relocation. Features thus emerge from our optimal transport procedure rather than from an explicit feature detection scheme. The resulting reconstruction is extracted from the simplified triangulation through edge filtering, guaranteeing an intersection-free output.

1.3 Outline

The thesis is organized as follows: Chapter 2 reviews the basics of optimal transport and introduces a novel formulation for the problem of reconstructing and simplifying 2D point sets; Chapter 3 describes the algorithm and its implementation details; Chapter 4 presents examples of reconstructed shapes; and Chapter 5 discusses limitations and future work.

Chapter 2

Foundations

We first review the basics of optimal transport before providing details on how we approach the problem of reconstructing and simplifying 2D point sets using the optimal transport framework.

2.1 Basics of Optimal Transport

Optimal transport refers to the problem of optimizing the cost of transportation and allocation of resources [40]. An intuitive example of optimal transport (used initially by Gaspard Monge in 1781) consists in determining the most effective way to move a pile of sand to a hole of the same volume—"most effective" here meaning that the integral of the distances the sand is moved (one infinitesimal unit of volume at a time) is minimal. This formulation of the problem is referred to as Monge's variational formulation and assumes that the sand is moved through a *point-to-point* mapping called the *transport plan*.

Defining a transport plan as an injective map, however, raises difficulties in the construction of optimal plans. This issues occurs even in simple examples such as the transportation of mass from a point to a uniform distribution on a line segment: since the dimensions of the support domains differ, the transport plan from a point to a segment is ill-defined.

In order to relax this restriction on transport plans, Kantorovich extended the formulation to deal with transport plans between two probability measures μ and ν. In this formulation, a transport plan is a probability measure π on support$(\mu) \times$ support(ν) whose marginals are μ and ν. If μ and ν have finite support, $\pi(x, y)$ specifies the amount of mass transferred from x to y.

For any such transport plan we can associate a notion of cost. A common cost function is the q-Wasserstein metric w_q, defined as

$$w_q(\mu, \nu) = \left(\inf_{\pi} \int_{\mathbb{R}^d \times \mathbb{R}^d} \| x - y \|^q \ \mathrm{d}\pi(x, y) \right)^{1/q}.$$

To reuse the example mentioned above, if each distribution is viewed as a unit amount of piled-up sand, this metric is the minimum L_q distance of turning one pile into the other.

So far we have assumed that the measures μ and ν are probability distributions, hence both with unit total mass. Throughout this work, we adopt a trivial extension of the w_2-distance that incorporates the total amount of mass to be transported. More specifically, for any pair of measures μ and ν with bounded support in $\mathbb{R}^2$ and common total mass M, we define the optimal transport cost as

$$W_2(\mu, \nu) = \sqrt{M}\ w_2(\mu/M, \nu/M). \tag{2.1}$$

2.1.1 Applications

Optimal transport is a relevant topic in many fields [40]. Perhaps, its most notorious application is the problem of facility location in economics and operational research. In computer graphics, the optimal transport theory has been recently applied to geometry processing tasks such as shape inference and registration. Chazal et al. [8], for example, exploited the optimal transport theory to design approximations of unsigned distance functions which are robust to noise and outliers. Mullen et al. [34] leveraged such distance functions and proposed a robust method to reconstruct smooth surfaces from raw point sets. Lipman and Daubechies [29], on the other hand, presented a solution for the transportation problem between conformal mass densities in order to determine similarities between 3D shapes. In computer vision, optimal transport has also been used as a means to compare [37], transfer [11] and quantize colors in images [6]. Physical simulations can also benefit from optimal transportation as shown in [27, 28] for the specific problems of ballistic impact simulations and fluid dynamics. At last, Mullen et al. [35] used optimal transport to derive tight bounds on the accuracy of discrete differential operators and then formulated a family of functionals for meshing optimization.

2.2 Formulation

We address the reconstruction of a point set $\mathcal{S}$ by approximating $\mathcal{S}$ with the vertices and edges of a *triangulation* $\mathcal{T}$. To this end, we regard the point set $\mathcal{S}$ and the vertices and edges of $\mathcal{T}$ as measures.

More specifically, each input point $p_i \in \mathcal{S}$ is seen as a Dirac measure μ_i centered at p_i and of mass m_i. The point set is thus considered as a measure $\mu = \sum_i \mu_i$ with total mass $M = \sum_i m_i$. We then associate each 0- and 1-simplex of $\mathcal{T}$ to a uniform measure supported over the simplex itself; that is, a vertex v is also seen as a Dirac measure, but an edge e is a uniform 1D measure defined over a line segment. We further call M_v and M_e the mass of a vertex v and an edge e, respectively, such that:

$$\sum_{v \in \mathcal{T}} M_v + \sum_{e \in \mathcal{T}} M_e = M. \tag{2.2}$$

Our general approach to reconstruct and simplify a point set $\mathcal{S}$ can thus be phrased as follows:

Considering $\mathcal{S}$ as a measure μ consisting of Dirac masses, find a coarse simplicial complex $\mathcal{T}$ such that μ is well approximated in terms of the W_2 cost by a linear combination of uniform measures on the edges and vertices of $\mathcal{T}$.

2.2.1 Benefits

The main advantage of the notion of optimal transport distance in the context of reconstruction and simplification is its robustness to noise and outliers [8]. Moreover, it captures two complementary notions of approximation between $\mathcal{S}$ and $\mathcal{T}$. First, it measures a symmetric approximation error between shapes since $W_2(\mu, \nu) = W_2(\nu, \mu)$, avoiding the notorious shortcomings of asymmetric distances. Second, the optimal transport cost measures the local defect of uniform sampling of the point set along the reconstructed edges; minimizing such a metric favors edges covering uniformly dense regions while preserving boundaries and sharp features (Fig 2.1).

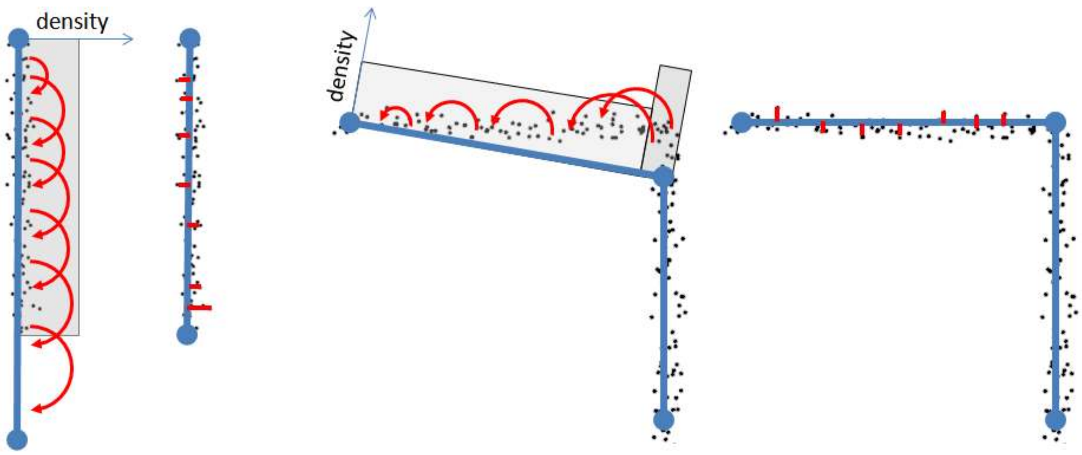


Figure 2.1: *Boundary and feature preservation.* Left: *An edge extending beyond (or falling short of) the point set would incur a large tangential component, enforcing preservation of boundaries.* Right: *When an edge does not fit a sharp corner, the transport cost has both a large tangential component (due to the non-uniform sampling along the edge) and a large normal error. Minimizing transport cost will thus induce alignment of edges to features.*

2.3 Transport Plan & Cost from Points to Simplices

In order to compute the approximation error between $\mathcal{S}$ and $\mathcal{T}$, we need to define the plan mapping $\mathcal{S}$ to $\mathcal{T}$ such that optimizes the W_2 cost between their respective measures. Let us assume for now that we are given a triangulation $\mathcal{T}$, and an *point-to-simplex assignment* which maps every input point p_i to either an edge e or a vertex v of $\mathcal{T}$—we will explain in Sec. 2.4 how this assignment is automatically determined. Equipped with this point-to-simplex assignment, we can derive an *optimal transport* π from $\mathcal{S}$ to vertices and edges of a triangulation $\mathcal{T}$.

In the following, we consider an arbitrary vertex v and an arbitrary edge e along with their respective assigned points from $\mathcal{S}$: we denote by $\mathcal{S}_v$ the set of points assigned to the vertex v and by $\mathcal{S}_e$ the set of points assigned to the edge e. Note that we assume these sets to be disjoint, with $\bigcup_{v\in\mathcal{T}} \mathcal{S}_v \cup \bigcup_{e\in\mathcal{T}} \mathcal{S}_e = \mathcal{S}$. This way, we hold Eq. (2.2) by setting M_v and M_e to the total mass of $\mathcal{S}_v$ and $\mathcal{S}_e$, respectively. Finally, we denote by π the optimal plan satisfying the prescribed assignment from points to simplices, and $W_2(\pi)$ its (optimal) cost.

2.3.1 Points to Vertex

For a vertex v, the optimal plan to transport the measure $\mathcal{S}_v$ to the Dirac measure centered on v with mass M_v is trivial, and its local W_2 cost easily computed as

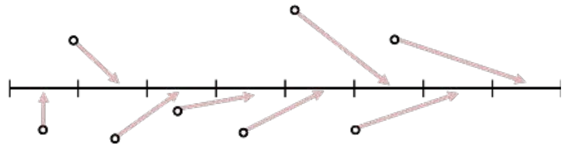
$$W_2(v, \mathcal{S}_v) = \sqrt{\sum_{p_i\in\mathcal{S}_v} m_i \|p_i - v\|^2}. \tag{2.3}$$

2.3.2 Points to Edge

For an edge e, the optimal plan (and its associated W_2 cost) is less straightforward to express as it requires transporting a mass M_e from $\mathcal{S}_e$ to a *uniform* measure of value $M_e/|e|$ defined over the edge e, where $|e|$ denotes the edge length—notice that this edge measure integrates to M_e, ensuring the existence of a transport plan. Because our transport cost is based on the L_2 distance, we can decompose the optimal transport plan into a *normal* and a *tangential* component to e. The optimal normal plan is a simple orthogonal projection, and its transport cost N is expressed as

$$N(e, \mathcal{S}_e) = \sqrt{\sum_{p_i\in\mathcal{S}_e} m_i \|p_i - q_i\|^2}, \tag{2.4}$$

where q_i is the orthogonal projection of p_i onto e.



The tangential plan is slightly more involved to derive, but its cost can also be done in closed form. We proceed by first sorting the projected points $\{q_i\}$ along e and partitioning the edge into cardinality($\mathcal{S}_e$) segment bins, where the i-th bin length is $l_i = (m_i/M_e)|e|$. Each point q_i is then spread tangentially over the i-th bin so as to result in a uniform measure over e, defining the local optimal transport from $\mathcal{S}_e$ to e (see inset). Now consider a point p_i of mass m_i that projects onto q_i on edge e; set a 1D coordinate axis along the edge with origin at the center of the i-th bin, and call c_i the coordinate of q_i in this coordinate axis. The tangential cost t_i of p_i is then computed as the accumulated distribution resulting from the order-1 moment of all points on the i-th bin with respect to the projection q_i, yielding:

$$t_i = \frac{M_e}{|e|} \int_{-l_i/2}^{l_i/2} (x - c_i)^2 \, dx = m_i \left(\frac{l_i^2}{12} + c_i^2 \right).$$

We can now sum up the cost t_i for every point p_i in $\mathcal{S}_e$ to get the tangential component T of the optimal transport cost for an entire edge:

$$T(e, \mathcal{S}_e) = \sqrt{\sum_{p_i \in \mathcal{S}_e} m_i \left(\frac{l_i^2}{12} + c_i^2 \right)}. \tag{2.5}$$

Note that Eq. (2.5) is exact; it is thus stable under refinement, in the sense that we obtain the same cost if we split each point p_i of mass m_i into several points that sum up to m_i and transport them onto smaller bins whose lengths are function of the new masses.

Optimal Cost. The total W_2 cost to transport $\mathcal{S}$ to $\mathcal{T}$ through the transport plan π is thus conveniently written by summing the contributions of every edge and vertex of $\mathcal{T}$:

$$W_2(\pi) = \sqrt{\sum_{e \in \mathcal{T}} [N(e, \mathcal{S}_e)^2 + T(e, \mathcal{S}_e)^2] + \sum_{v \in \mathcal{T}} W_2(v, \mathcal{S}_v)^2}\,. \tag{2.6}$$

2.4 Point-to-Simplex Assignment

Given a triangulation $\mathcal{T}$, we still need to define an assignment of the input point set $\mathcal{S}$ to 0- and 1-simplices of $\mathcal{T}$ which minimizes the total cost of the transport plan π described above. Alas, solving this problem exactly is computationally infeasible since the combinatorial complexity of all possible assignments becomes intractable as the size of the problem increases. We propose a simple heuristic that constructs an assignment from $\mathcal{S}$ to edges and vertices of $\mathcal{T}$ in two steps. Each point p_i is first temporarily assigned to the closest edge of the simplicial complex, resulting in a partition of $\mathcal{S}$ into a disjoint union of subsets $\mathcal{S}_e$, where each $\mathcal{S}_e$ contains the input points nearer to e than to any other edge. We then go over each edge e, and consider the two following assignments: either (i) keep $\mathcal{S}_e$ assigned to e, or (ii) assign instead each point of $\mathcal{S}_e$ to its closest endpoint of e. Using Eqs. (2.3)-(2.5) we choose, out of these two possibilities, the local assignment leading to the *smallest* total cost. Once this simple test has been performed for each edge, the vertices v and edges e of $\mathcal{T}$ all have their assigned points (respectively, $\mathcal{S}_e$ and $\mathcal{S}_v$) from $\mathcal{S}$, and the optimal transport plan (and cost) for this assignment is thus fully determined as derived in Sec. 2.3.

Note that this assignment provides a *natural characterization of the edges of* $\mathcal{T}$: an edge e is called a *ghost* edge when $\mathcal{S}_e = \emptyset$, i.e., no input points are transported onto it; conversely, e is called a *solid* edge if it receives a nonzero amount of mass from the point set. Ghost edges are, in effect, not part of the reconstruction, and are only useful to define the current simplicial tiling of the domain (see Fig. 3.1, center-right).

Chapter 3

Algorithm

With the point-to-simplex assignment and its induced transport plan described in the previous chapter, we now devise an algorithm to produce a coarse reconstruction $\mathcal{T}$ from the input point set.

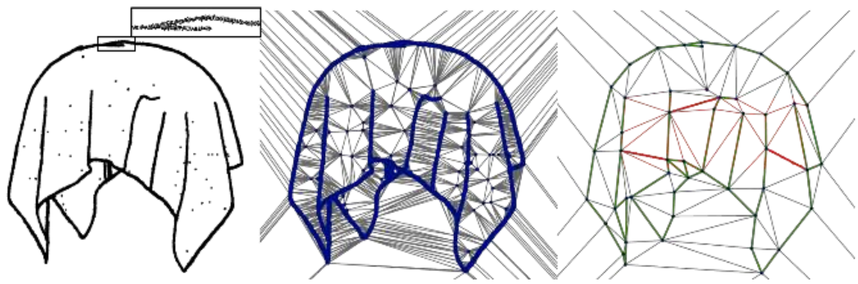


Figure 3.1: ***Algorithm Pipeline.*** From left to right: *input point set; Delaunay triangulation of input; after simplification, with ghost edges in grey, relevant solid edges in green, discarded solid edges in red; final reconstruction.*

3.1 Overview

The algorithm proceeds in a fine-to-coarse manner through greedy simplification of a Delaunay triangulation initialized from the input points. Simplification is performed through a series of half-edge collapse operations, ordered so as to minimize the increase of the total transport cost between the input point set and the triangulation. Vertices are also locally optimized after each edge collapse in order to further optimize the local assignment. Finally, the reconstruction is derived through edge filtering of the simplified mesh. We depict the pipeline in Fig. 3.1 and give pseudocode in Fig. 3.2.

// TRANSPORT-BASED RECONSTRUCTION
Input: point set $\mathcal{S} = \{p_1, \ldots, p_n\}$.

// *Initialization* (Sec. 3.2)
Construct Delaunay triangulation $\mathcal{T}_0$ of $\mathcal{S}$.
Compute initial plan π_0 from $\mathcal{S}$ to $\mathcal{T}_0$
$k \leftarrow 0$
repeat
  Pick best half-edge $e = (x_i, x_j)$ to collapse
  Set $\mathcal{N}_{i,m}$ the m-ring of x_i
  // *Simplification* (Sec. 3.3)
  Create $\mathcal{T}_{k+1}$ by merging x_i onto x_j
  // *Update Transport* (Sec. 2.4)
  $\pi'_{k+1} := \pi_k$ with local reassignments in $\mathcal{N}_{i,1}$
  // *Relocation* (Sec. 3.4)
  Optimize position of vertices in $\mathcal{N}_{i,1}$
  // *Update Transport* (Sec. 2.4)
  $\pi_{k+1} := \pi'_{k+1}$ with local reassignments in $\mathcal{N}_{i,2}$
  $k \leftarrow k + 1$
until (desired vertex count)
// *Final Extraction* (Sec. 3.5)
Filter edges based on relevance (optional)

Output: vertices and edges of $\mathcal{T}_n$.

Figure 3.2: *Pseudocode of the algorithm.*

3.2 Initialization

Our algorithm begins by constructing a 2D Delaunay triangulation $\mathcal{T}_0$ of the input point set $\mathcal{S}$. The resulting triangulation is augmented with four vertices placed at the corners of a loose bounding box of the input points; these vertices are not eligible for simplification as they act as pins ensuring that the sample points fall within the convex hull of $\mathcal{T}_0$. With this triangulation is associated a trivial optimal plan π_0, for which each point is assigned to its corresponding vertex in $\mathcal{T}_0$ for a total transport cost of zero.

3.3 Simplification

Simplification of the initial mesh $\mathcal{T}_0$ is performed through half-edge collapses, each one removing one vertex and three edges from the triangulation (Fig. 3.3). As a collapse turns the current triangulation $\mathcal{T}_k$ into a new triangulation $\mathcal{T}_{k+1}$, it induces a local change of the transport plan π_k between $\mathcal{S}$ and $\mathcal{T}_k$ and, consequently, a change of the total cost $W_2(\pi_k)$ (denoted δ hereafter, with $\delta_k = W_2(\pi_{k+1}) - W_2(\pi_k)$).

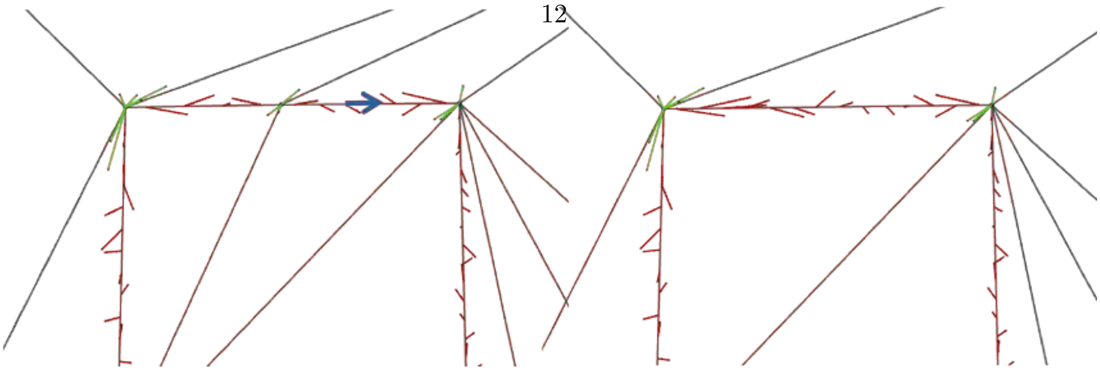


Figure 3.3: ***Edge simplification.*** *Half-edge collapse (arrow) triggers a local reassignment of the input points (assignments in red for edges, green for vertices).*

As our goal is to minimize the increase in total cost, we apply collapses in *increasing order* of δ. To this end, we initially simulate all feasible collapses to determine their associated δ and fill up a dynamic priority queue sorted in increasing order of δ. Decimation is then achieved by repeatedly popping from the queue the next edge to collapse, performing the collapse, updating the transport plan and cost according to Sec. 2.4, and updating the priority queue. Since our approximation of the transport plan partitions the input points based on their respective nearest edge, updating the transport involves only the edges confined to the one-ring of the removed vertex of the edge collapse (Fig. 3.4, center). Updating the priority queue, however, is required within a larger stencil around the removed vertex: edges incident to the modified one-ring and emanating from its flap vertices have their respective δ impacted by the new point-to-simplex assignment (Fig. 3.4, right).

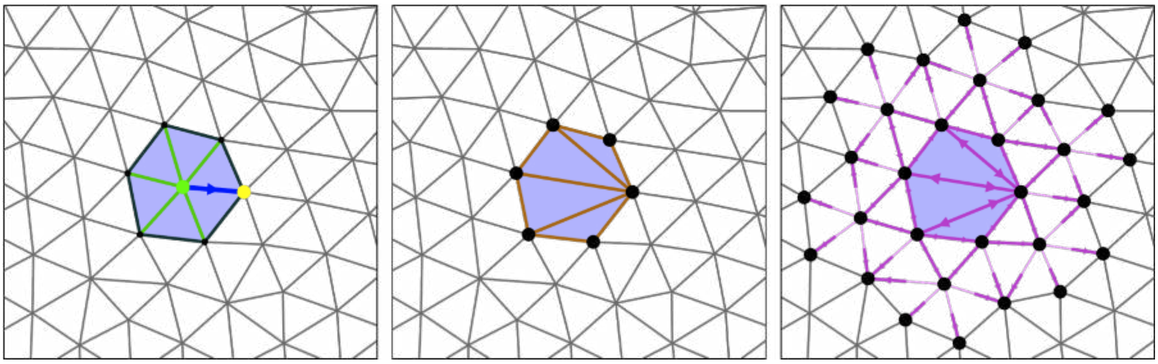


Figure 3.4: ***Updates during collapse.*** Left: *collapse operator.* Middle: *edges with updated cost.* Right: *collapse operators pushed to the priority queue.*

3.3.1 Systematically Collapsing Edges

A half-edge is said to be *collapsible* if its collapse creates neither overlaps nor fold-overs in the triangulation. In order to preserve the embedding of the triangulation, we must only target collapsible edges during simplification. This is achieved by verifying both topological and geometric conditions: the former corresponds to the so-called *link condition* [12]; the latter consists of checking whether the target vertex of the half-edge is within the *kernel* [36] of the polygon formed by the one-ring of the source vertex (Fig. 3.5, left). While simple and commonly used, these validity conditions invalidate around 30% of the candidate collapse operators during the complete course of a simplification. This can severely affect the performance of our greedy approach to optimal transport as it relies on targeting the edge with the least cost first. To overcome this issue and ensure that the greedy decimation is systematically processed in increasing order of cost, we modify the collapse operator through a local edge flip procedure that makes *every* edge collapsible as we now review.

Suppose we want to collapse a half-edge (x_i, x_j). We denote by P_{x_i} the (counter-clockwise oriented) polygon formed by the one-ring of x_i, and by K_{x_i} its kernel. We say that an edge $(a, b) \in P_{x_i}$ is *blocking* x_j if the triangle (x_j, a, b) has clockwise orientation. We also call an edge *flippable* if its endpoints and its two opposite vertices form a convex quadrilateral. The idea of the flipping procedure is to elongate K_{x_i} in the direction of (x_i, x_j) until it includes x_j. For every blocking edge $(a, b) \in P_{x_i}$, we call D as the distance from x_j to the intersection point between the supporting lines of (x_i, x_j) and (a, b). This distance indicates how much K_{x_i} can be elongated by removing (a, b) from P_{x_i}. We then sort the blocking edges in a priority queue in decreasing order of distance D, and select the top edge to be removed from P_{x_i}. We remove an edge (a, b) from P_{x_i} by flipping either the edge (a, x_i) or the edge (b, x_i). Note that, since (a, b) has the greatest distance D within P_{x_i}, either (a, x_i) or (b, x_i) is flippable, otherwise (a, b) would not have been at the top of the queue. Between these two choices, we pick the edge with the smallest D when flipped, as it elongates K_{x_i} the most. By repeating edge flips, we iteratively move K_{x_i} towards x_j and this procedure is guaranteed to terminate when the modified P_{x_i} no longer has blocking edges (Fig. 3.5). It is worth pointing out that x_i may not be inside the final P_{x_i} (Fig. 3.5, bottom); however, since the flipping procedure is always associated to a half-edge collapse, x_i is later deleted by the collapse and thus the mesh is always kept valid.

3.4 Vertex Relocation

The simplification scheme presented so far restricts the successive triangulations $\{\mathcal{T}_k\}$ obtained through simplification to have vertices on input points. However, noise and missing data usually preempt having the exact location of sharp corners among the input points, making interpolated triangulations poorly adapted to recover features. In order to better preserve features, we perform a vertex relocation (in the same spirit as [20]) after each half-edge collapse (Fig. 3.6).

The vertex relocation is designed to improve the fitting of vertices and edges of the triangulation to the input data: we move vertices in order to further minimize the *normal* component of the current

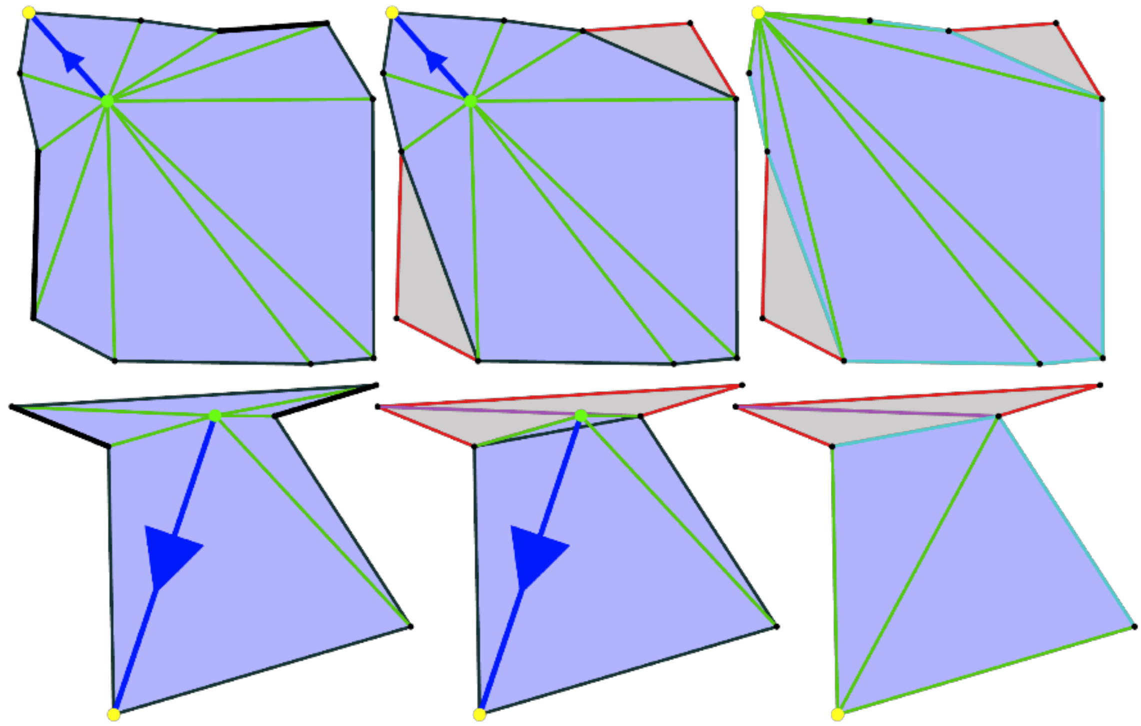


Figure 3.5: ***Making edges collapsible.*** Left: *collapsing the edge (in blue) creates fold-overs because of blocking edges (in black).* Middle: *the flipping procedure makes the edge collapsible.* Right: *after collapse.*

W_2 distance (defined in Eq. (2.3) and Eq. (2.4)). Remembering that the square of the normal part of the W_2 cost associated with a vertex v of $\mathcal{T}$ can be written as

$$\sum_{p_i \in \mathcal{S}_v} m_i \|p_i - v\|^2 + \sum_{b \in \mathcal{N}_1(v)} \sum_{p_i \in \mathcal{S}_{(v,b)}} m_i \|p_i - q_i\|^2,$$

we compute the optimal position v^* of v by equating the gradient of the above expression to zero. If we denote by λ_i the barycentric coordinate of the projection q_i of the input point p_i onto an edge (v, b), i.e., $q_i = (1 - \lambda_i)v + \lambda_i b$, then the optimal position is given by:

$$v^* = \frac{\sum_{p_i \in \mathcal{S}_v} m_i p_i + \sum_{b \in \mathcal{N}_1(v)} \sum_{p_i \in \mathcal{S}_{(v,b)}} m_i (1 - \lambda_i)(p_i - \lambda_i b)}{M_v + \sum_{b \in \mathcal{N}_1(v)} \sum_{p_i \in \mathcal{S}_{(v,b)}} m_i (1 - \lambda_i)^2}.$$

In practice, we move the vertex v to v^* only if the resulting triangulation $\mathcal{T}_{k+1}$ is still an embedding (i.e., v^* is inside the kernel of the one-ring of v). Once this relocation is achieved, we proceed as we did for edge collapses: we collect the input points affected by this relocation, assign them to their nearest edge, and determine the new transport plan π_{k+1}.

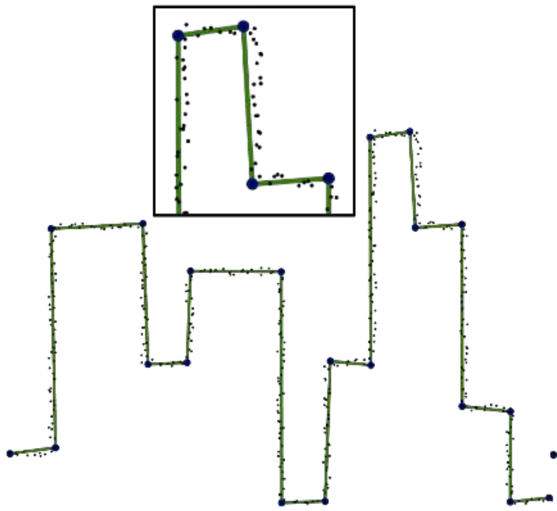


Figure 3.6: ***Vertex relocation.*** *Noisy skyline point set, before (left) and after (right) vertex relocation.*

3.5 Edge Filtering

The triangulation resulting from repeated decimation provides a tessellation of the domain that approximates the inferred shape by a subset of its edges. As described in Sec. 2.4, we already know whether an edge is relevant to the shape: only *solid* edges carry mass from the input point set. However, the presence of noise, outliers, and scale-dependent features may still lead to a few undesirable solid edges (Fig. 3.1, center-right in red). While the final edge filtering is most likely quite application-dependent and often unnecessary, we found it satisfactory to offer the option to eliminate solid edges based on a notion of relevance r_e, with

$$r_e = \left(\frac{|e|}{w_2(e, \mathcal{S}_e)}\right)^2 = \frac{M_e |e|^2}{N(e, \mathcal{S}_e)^2 + T(e, \mathcal{S}_e)^2}.$$

Observe that r_e is inversely proportional to the w_2-density cost over the edge e and, therefore, indicates more importance for long edges with uniform sampling distribution. In our experiments, a threshold on relevance was only applied to examples with large amount of outliers (Figs. 3.1, 4.3, and 4.6).

3.6 Implementation details

We now describe a few implementation details which improve the efficiency and flexibility of the algorithm.

3.6.1 Initialization

For large datasets, a significant speed-up can be obtained by using only a random subset of the input points for $\mathcal{T}_0$. This trivial efficiency improvement does not lead to visible artifacts as we optimize the vertices of subsequent triangulations through our relocation procedure.

3.6.2 Multiple-Choice Priority Queue

The reconstruction of large datasets requires the simulation of thousands of collapse operations, and thus, our algorithm initially spends a significant amount of time performing frequent updates of the dynamic priority queue. We can speed up the simplification process through a multiple-choice approach [41] without significant degradation of our results: for each atomic simplification step we simulate only a random subset of the half-edges (typically 10) and collapse the half-edge from this subset with the smallest δ. When the mesh has 5 times the targeted number of vertices or less, we switch back to an exhaustive priority queue as the order of collapse for coarse meshes can significantly affect the final results. With this simple strategy, a five- to six-time speedup is typically obtained (see Tab. 4.1).

3.6.3 Postponing Vertex Relocation

Vertex relocation is effective only when a sufficient number of input points are assigned to each edge of the triangulation. Obviously, this number is very small at the beginning of the algorithm and increases as the triangulation becomes coarser. For this reason, we turn off the vertex relocation step during most of the simplification process, activating it only for the last 100 decimations. This strategy to lower computational cost also has little to no effect on the final results.

3.6.4 Reconstruction Tolerance

It is sometimes desirable to control the maximum error induced by the reconstruction, especially in the case of outlier-free point sets. We add an option which allows discarding from the priority queue each collapse operator which would induce a transport with maximum normal and/or tangential cost above a user-specified tolerance. The algorithm terminates when the priority queue is empty (Fig. 3.7 is the only figure where this option was activated).

Figure 3.7: ***Tolerance Control.*** *A maximum normal and tangential tolerance error (tolerance volume in gray, reconstruction in green) are imposed during simplification on the shark model.*

Chapter 4

Results

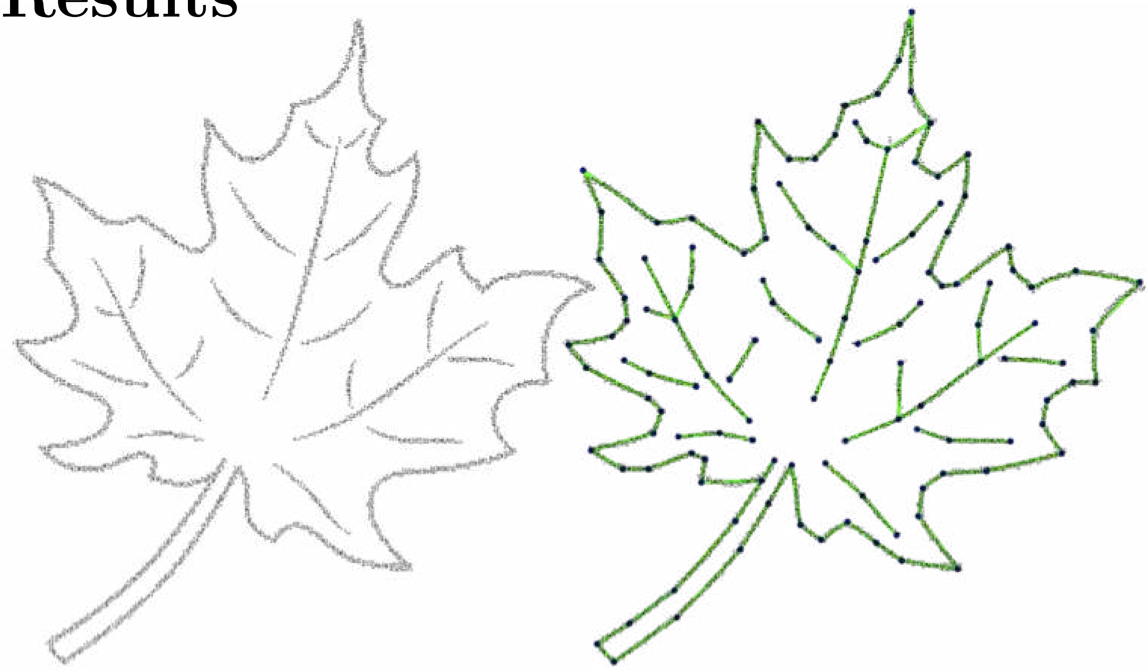


Figure 4.1: ***Leaf.*** *Example with sharp corners, boundaries, and branching points.*

We implemented our reconstruction and simplification algorithm using the CGAL library [7]. It takes as input point sets, possibly with mass attributes, and a desired vertex count of the final reconstruction. Timings for typical examples computed on 2.4GHz Intel Core 2 Duo with 2GB RAM are given in the following table.

Data	Points	EX (s)	MC (s)
Shark	440	2.6	0.5
Bird	800	4.6	0.7
Horse	1200	6	0.9
Star	3k	18	3.6
Australia	5k	39	7
Falcon	8k	70	15
Table cloth	20k	248	51

Table 4.1: ***Timing.*** From left to right: *Data set, number of points, reconstruction time in seconds using an exhaustive priority queue (EX) vs. a multiple choice (MC) decimation strategy.*

On closed and noise-free curves, our approach performs comparably with state-of-the-art reconstruction and simplification methods applied in sequence. However, our approach is noticeably better at dealing with cases where the point set contains noise, outliers, and features. For instance, combinatorial methods such as Crust [21] fail to properly reconstruct a point set as soon as noise is added (Fig. 4.2); Poisson-like implicit approaches [23] have also been reported to fail in this case unless reliable normals are provided [2]. In contrast, our transport-based method is robust to large amounts of noise and outliers as illustrated in Fig. 4.3 where more outliers than initial points still leads to a good reconstruction—and *sharp features are still captured* (see also Fig. 4.5) without the typical blunting that most previous work (based on local denoising) would create. Fig. 4.10 and 4.9 depict how the reconstruction gracefully degrades when either the vertex count or the point density decreases. Our method also handles complex shapes with branching, intersections, and open curves without additional parameters to tweak (see the leaf example in Fig. 4.1).

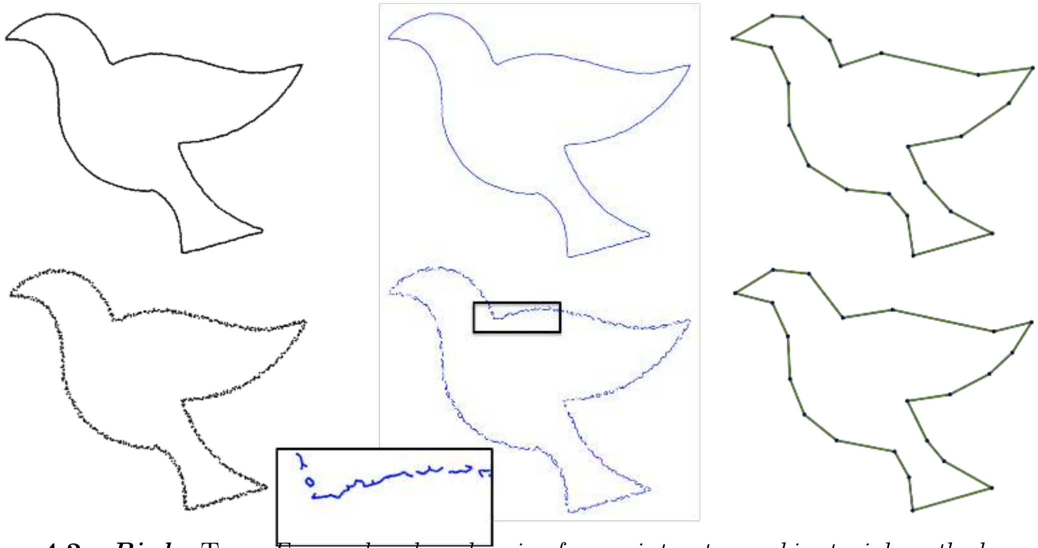


Figure 4.2: ***Bird.*** Top: *For a closed and noise-free point set, combinatorial methods such as crust [21] finds a connectivity for all input points, while we obtain a simplified reconstruction.* Bottom: *When the point set is hampered with synthetic noise (*0.5% *of bounding box), combinatorial approaches fails to recover the shape. Our method is noise-resilient and returns a result similar to the noise-free case.*

Our approach performs equally well on unprocessed, non-synthetic datasets. Images of line drawing, for instance, are particularly challenging for image processing techniques to approximate with polygonal curves. Nevertheless, our method is able to robustly extract such shapes by considering the gray level pixels as points with a mass proportional to the pixel intensity. For instance, Fig. 4.7 depicts the reconstruction of a complex mechanical illustration (a comparison to a commercial image vectorization product is provided), and Fig. 4.8 shows the vectorization of a scanned artist drawing. We also show how our method performs on GIS data in Fig. 4.4, comparing our results with the well known Douglas-Peucker algorithm [22] and a direct implementation of the Garland-Heckbert simplification approach [20].

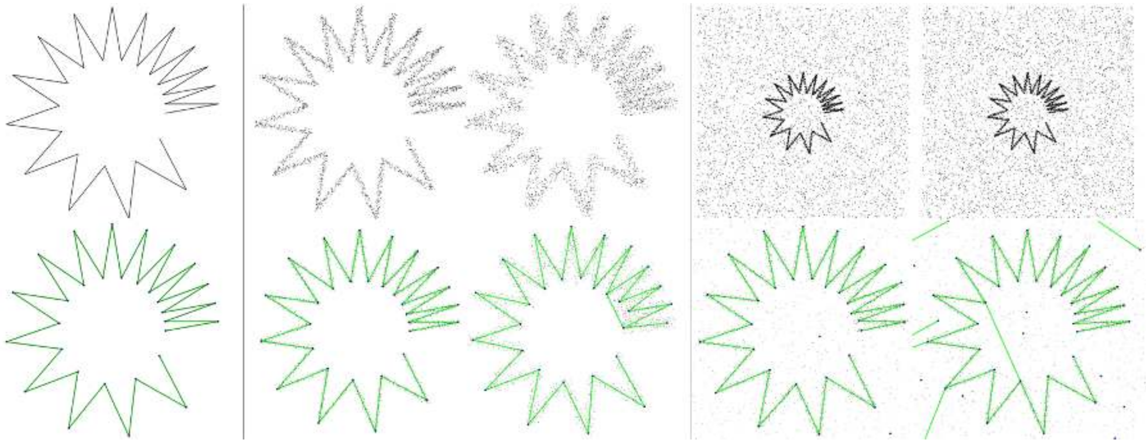


Figure 4.3: ***Robustness to noise and outliers.*** *The input shape (3K points) has sharp corners subtending small angles as well as boundaries. Our reconstruction is perfect for a noise-free input (left); as noise is added (middle, 2% and 2.5% of bounding box), the output degrades gracefully, still capturing most of the sharp angles; even after adding 4K or 4.5K outliers and 2% of noise (right), the reconstruction remains of high quality, although artifacts start appearing in this regime.*

Fig.5.1(top) also illustrates how our approach infers shape at various scales, from fine to coarse. The original assignment for $\mathcal{T}_0$ implies a vertex discretization of the datasets as all edges are initially classified as ghost edges; then these vertices are regrouped in 1-simplices, first consolidating the reconstruction of highly sampled regions. Finally, as the triangulation is simplified, we are getting a coarse scale approximation of the input data. Note that on inputs that mix anisotropic and isotropic point densities (Fig. 4.6), we obtain mixed reconstructions composed of line segments and isolated vertices, which also matches the input well.

Finally, we show the versatility of our approach on a variety of datasets, varying from (kanji) characters, to scanned drawings, and even to gradient magnitudes obtained via finite-differences from an image (which provides a noisy sampling of the main contours of the image).

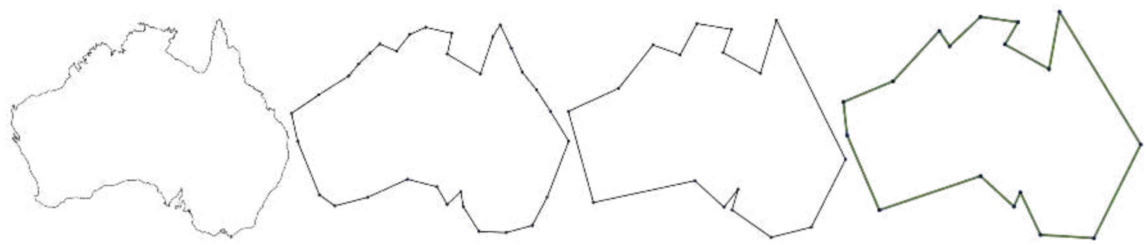


Figure 4.4: ***Australia Map.*** *We compare our approach (right) with two curve simplification algorithms: QEM [20] (left-center) and Douglas-Peucker [22] (center-right).*

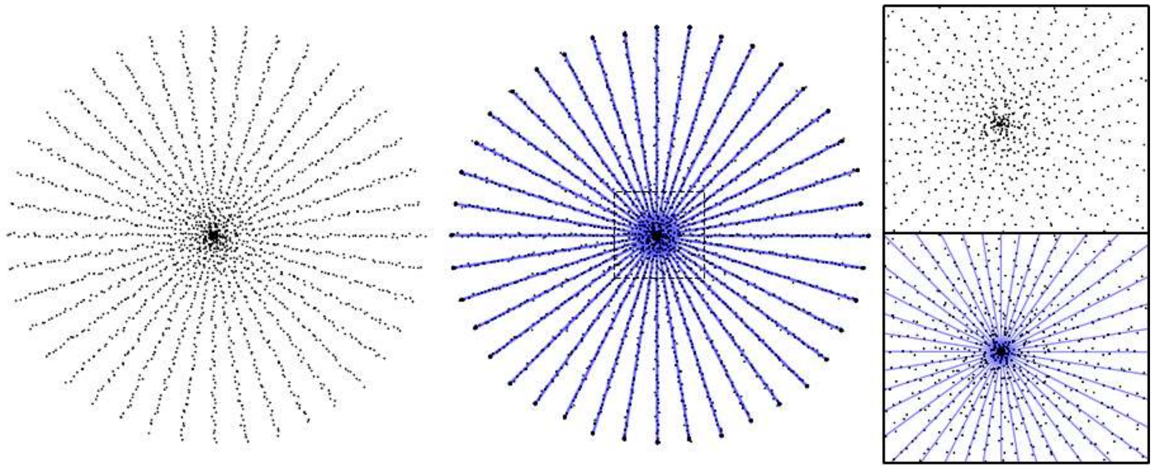


Figure 4.5: ***Noisy star.*** *The lines of the inferred shape near the star center are visually indistinguishable (see closeup). Nevertheless, our method reliably recovers the lines and intersection point by interleaving decimation and relocation.*

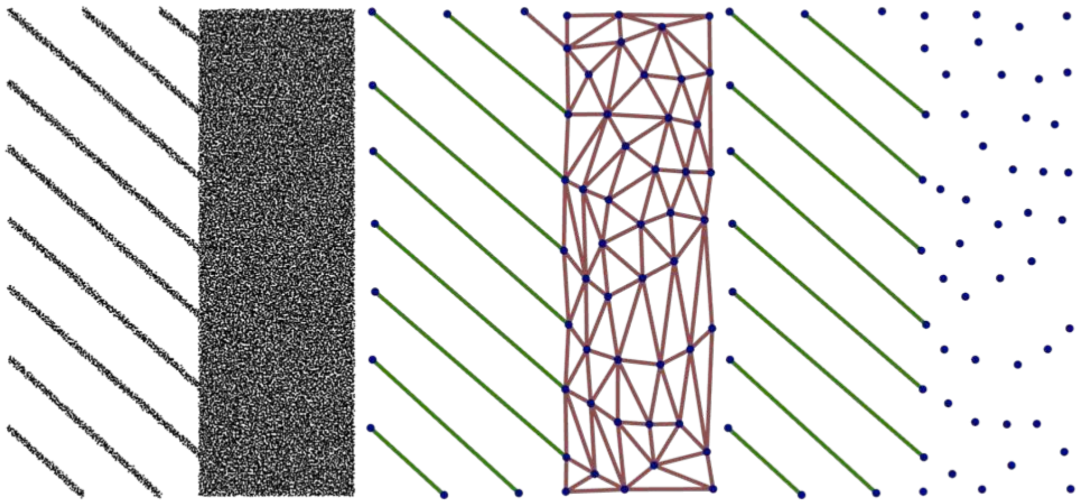


Figure 4.6: ***Mixed reconstruction.*** *By tuning the edge filtering we can extract a reconstruction made of either edges or of a mixture of edges and vertices.*

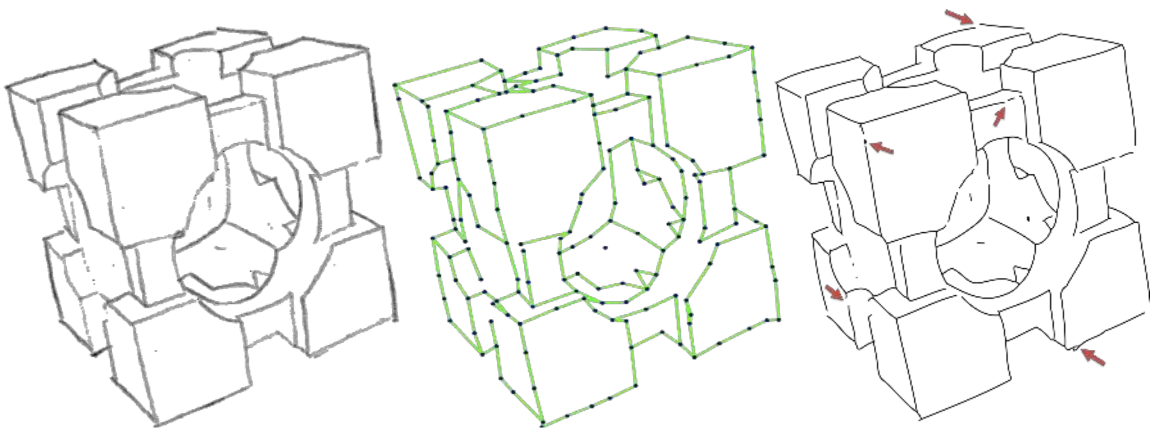


Figure 4.7: ***Mechanical part.*** *Line drawing with complex features (left). Our reconstruction (middle) preserves intersection, corners, and manages to even remove small wiggles and glitches. In comparison, Adobe's Live Trace® fails to handle glitches of this sketch (red arrows).*

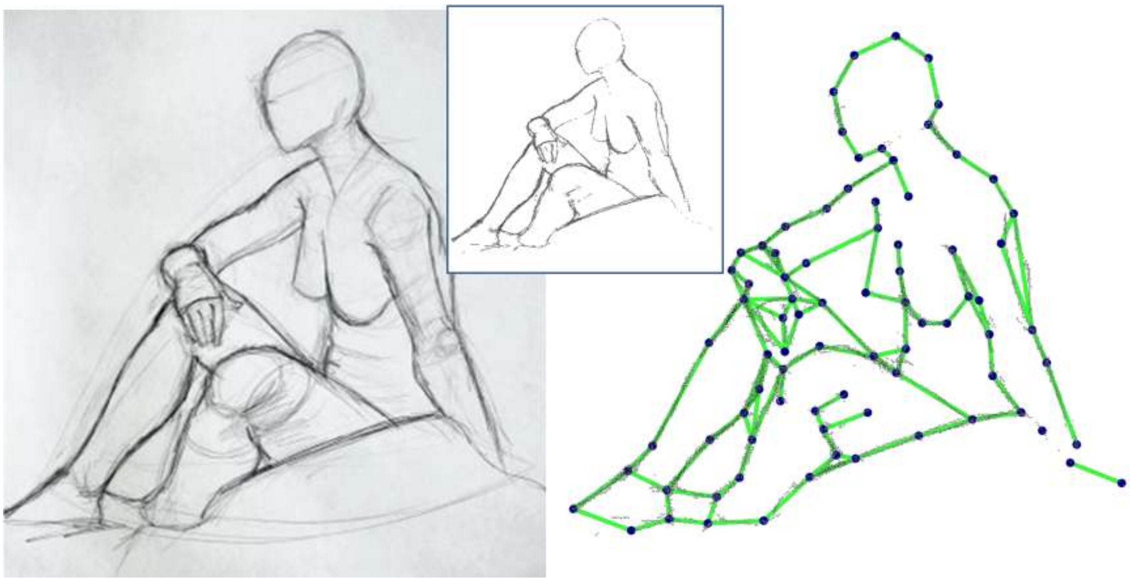


Figure 4.8: ***Woman.*** *Line drawing with variable thickness and outliers due to scanning artifacts (left). We apply thresholding on the input image (middle top) to reduce the number of input points before applying our reconstruction (right).*

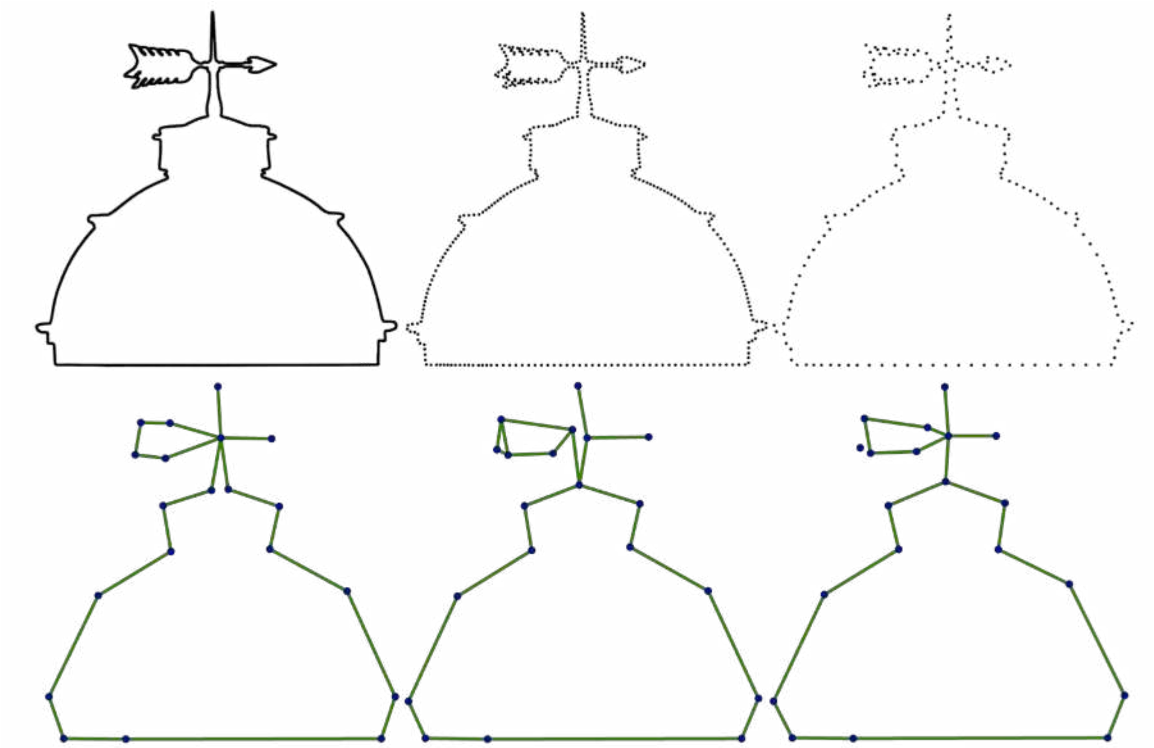


Figure 4.9: ***Light house.*** *Reconstructions with 20 vertices from input points with decreasing sampling density (from left to right: 2K, 400, and 200 points respectively).*

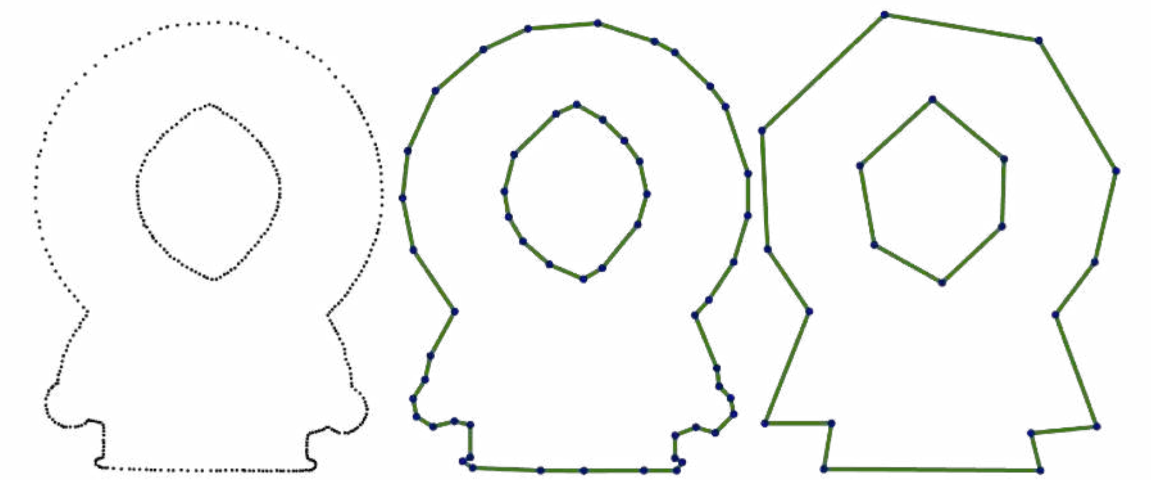


Figure 4.10: ***Kenny.*** *Reconstructions at different levels of simplification (54 and 20 vertices, respectively) from an input dataset of 364 points – data courtesy of South Park®.*

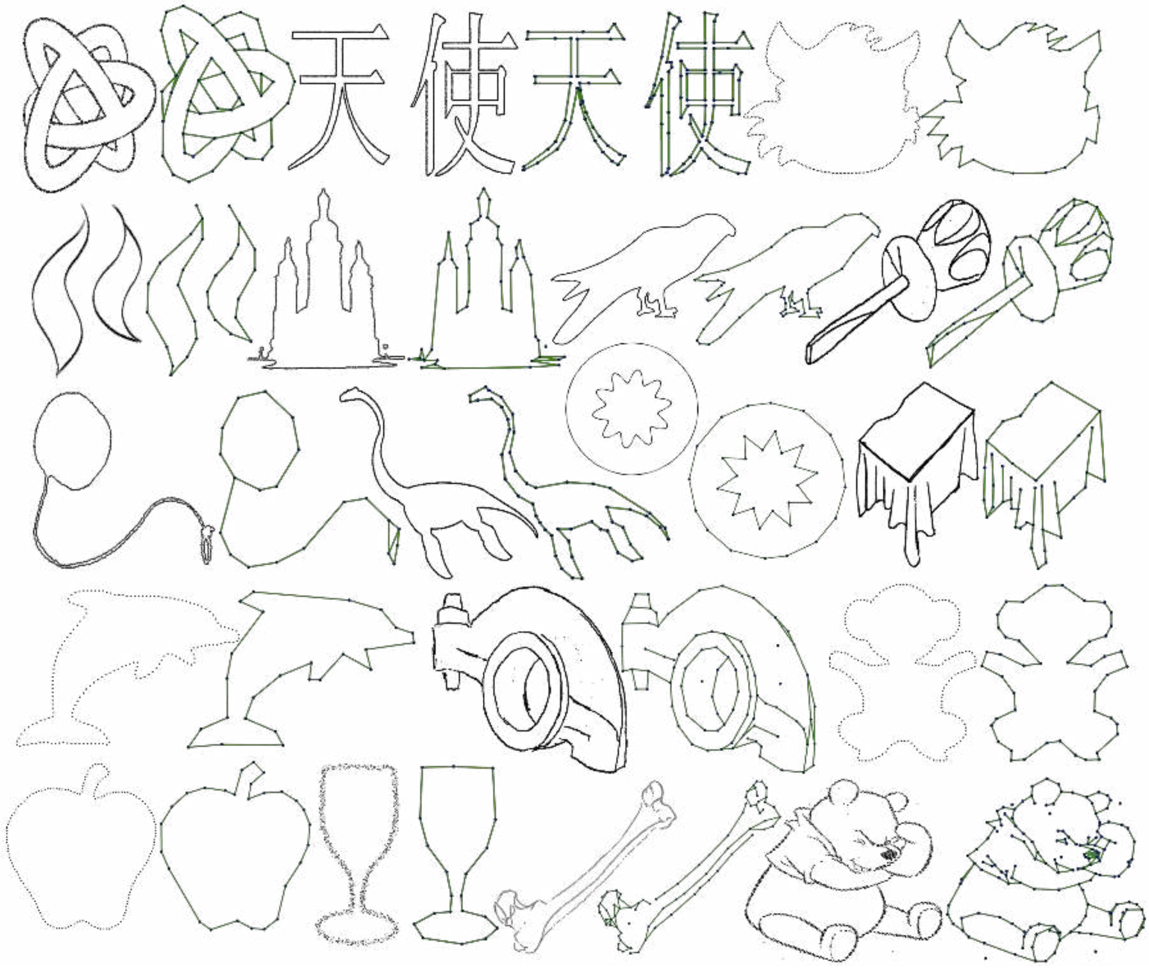


Figure 4.11: ***Gallery.*** *Plate of shapes reconstructed through our optimal transport driven approach with input points originating from noiseless dataset (cartoon face contour, circle with star, dolphin, monkey, apple), scanned drawings (screwdriver, tablecloth, rocketarm, bone), grey-level images (kanji characters, flames, Winnie the Pooh – data courtesy of Disney®), and image gradients computed by finite difference (all others).*

Chapter 5

Conclusion

We have presented a novel optimal transport formulation and derived a practical algorithm to tackle the problem of reconstruction and simplification of 2D shapes from unorganized point sets. We view the point set as a measure that we greedily approximate, in the optimal transport sense, by a sum of uniform measures on the vertices and edges of a simplicial complex.

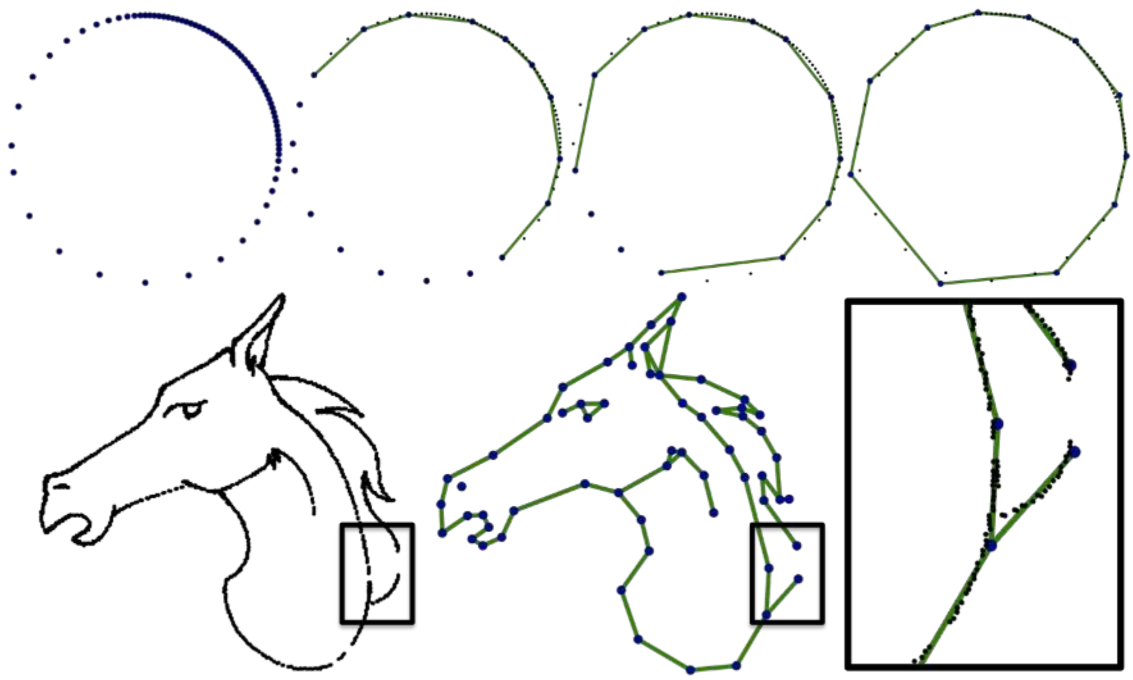


Figure 5.1: ***Sampling Density.*** Top: *the transport cost favors the decimation of areas with high point density. As we further simplify the triangulation, regions with lower densities are also recovered.* Bottom: *areas with missing data are reconstructed as boundaries.*

Strengths

The main value of our approach is the ability to robustly deal with feature preservation, such as sharp corners, cusps, intersections and boundaries. Another interesting property is the consistency between the underlying shape representation model (points and line segments) and the transport distance we considered: as this model inherently approximates features, we do not need any ad-hoc feature detection to preserve them.

Weaknesses

Despite satisfactory robustness to noise and outlier, our method does not deal gracefully with *widely* variable sampling. We see such examples in Fig. 5.1, where the reconstruction performs well on areas of high density but may fail to complete the undersampled area. This is quite natural given that our approach makes no assumption that the curve is closed, so undersampling can be interpreted as small features. As a consequence, our approach is also not resilient to missing data as the transport formulation seeks to preserve boundaries; however, very coarse reconstructions do recover the global shape quite well.

Future work

The problem of finding or approximating an optimal transport cost is of interest in many fields. In this work we have proposed a simple approximation strategy that assigns points to simplices in a local fashion, but more global alternatives (possibly through multiresolution) may be desirable. The simplification algorithm could also potentially benefit from a richer set of mesh decimation operators. For the final edge filtering step, we wish to incorporate application-dependent criteria so as to favor or enforce, for example, the reconstruction of 1-manifolds. Finally, the extension of this approach to robust surface reconstruction is probably the most exciting direction for future work.

Bibliography

[1] Pankaj K. Agarwal, Sariel Har-Peled, Nabil H. Mustafa, and Yusu Wang. Near-linear time approximation algorithms for curve simplification. *Lecture Notes in Computer Science*, 2461:29–41, 2002.

[2] P. Alliez, D. Cohen-Steiner, Y. Tong, and M. Desbrun. Voronoi-based variational reconstruction of unoriented point sets. In *Symposium on Geometry processing*, pages 39–48, 2007.

[3] Nina Amenta, Marshall Bern, and David Eppstein. The crust and the beta-skeleton: Combinatorial curve reconstruction. *Graphical Models and Image Processing*, 60(2):125–135, 1998.

[4] Dominique Attali. R-regular shape reconstruction from unorganized points. In *Proc. of ACM Symp. on Comp. Geometry*, pages 248–253, 1997.

[5] Haim Avron, Andrei Sharf, Chen Greif, and Daniel Cohen-Or. ℓ_1-sparse reconstruction of sharp point set surfaces. *ACM Trans. on Graphics*, 29(5):1–12, 2010.

[6] Michael Balzer, Thomas Schlömer, and Oliver Deussen. Capacity-constrained point distributions: a variant of lloyd's method. *ACM Trans. Graph.*, 28:86:1–86:8, July 2009.

[7] CGAL, Computational Geometry Algorithms Library. http://www.cgal.org.

[8] F. Chazal, D. Cohen-Steiner, and Q. Mérigot. Geometric inference for measures based on distance functions. Technical Report 00383685, INRIA, 2009.

[9] Siu-Wing Cheng, Stefan Funke, Mordecai Golin, Piyush Kumar, Sheung-Hung Poon, and Edgar Ramos. Curve reconstruction from noisy samples. In *Proc. of the Conf. on Comp. Geometry*, pages 302–311, 2003.

[10] F. de Goes, S. Goldenstein, M. Desbrun, and L. Velho. Exoskeleton: Curve network abstraction for 3d shapes. *Computers & Graphics*, 35(1):112 – 121, 2011.

[11] Julie Delon, Julien Salomon, and Andrei Sobolevski. Fast transport optimization for monge costs on the circle. *SIAM Journal of Applied Mathematics*, 70(7):2239–2258, 2010.

[12] T. K. Dey, H. Edelsbrunner, S. Guha, and D. V. Nekhayev. Topology preserving edge contraction. *Publ. Inst. Math. (Beograd)*, 66:23–45, 1999.

[13] Tamal K. Dey and Piyush Kumar. A simple provable algorithm for curve reconstruction. In *Proceedings of the Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 893–894, 1999.

[14] Tamal K. Dey, Kurt Mehlhorn, and Edgar A. Ramos. Curve reconstruction: Connecting dots with good reason. *Procedings of the 15th Annual ACM Symposion on Computational Geometry*, 15:229–244, 1999.

[15] Tamal K. Dey and Rephael Wenger. Fast reconstruction of curves with sharp corners. *Int. J. Comput. Geometry Appl*, 12(5):353–400, 2002.

[16] Christopher Dyken, Morten Dæhlen, and Thomas Sevaldrud. Simultaneous curve simplification. *Journal of Geographical Systems*, 11(3):273–289, 2009.

[17] H. Edelsbrunner, D. G. Kirkpatrick, and R. Seidel. On the shape of a set of points in the plane. *IEEE Trans. on Information Theory*, 29:551–559, 1983.

[18] S. Fleishman, D. Cohen-Or, and C. Silva. Robust moving least-squares fitting with sharp features. In *Proceedings of ACM SIGGRAPH*, 2005.

[19] Stefan Funke and Edgar A. Ramos. Reconstructing a collection of curves with corners and endpoints. In *Proceedings of the Symposium on Discrete Algorithms*, pages 344–353, 2001.

[20] Michael Garland and Yuan Zhou. Quadric-based simplification in any dimension. *ACM Transactions on Graphics*, 24(2):209–239, 2005.

[21] Christopher Gold and Jack Snoeyink. A one-step crust and skeleton extraction algorithm. *Algorithmica*, 30, 2001.

[22] John Hershberger and Jack Snoeyink. An $O(nlogn)$ implementation of the douglas-peucker algorithm for line simplification. In *Proceedings of Annual Symposium on Computational Geometry*, pages 383–384, 1994.

[23] Michael Kazhdan, M. Bolitho, and Hugues Hoppe. Poisson Surface Reconstruction. In *Proc. of SGP '06*, pages 61–70, 2006.

[24] Ravikrishna Kolluri, Jonathan R. Shewchuk, and James F. O'Brien. Spectral surface reconstruction from noisy point clouds. In *Proc. of SGP '04*, pages 11–21, 2004.

[25] Dmitry N. Krasnoshchekov and Valentin Polishchuk. Robust curve reconstruction with k-order alpha-shapes. In *Shape Modeling International*, pages 279–280, 2008.

[26] In-Kwon Lee. Curve reconstruction from unorganized points. *Computer Aided Geometric Design*, 17(2):161–177, 2000.

[27] B. Li, F. Habbal, and M. Ortiz. Optimal transportation meshfree approximation schemes for fluid and plastic flows. *International Journal for Numerical Methods in Engineering*, 2010.

[28] B. Li, A. Kidane, G. Ravichandran, and M. Ortiz. The optimal transportation meshfree (otm) method for ballistic limit impact simulations. In *Proc. of the IMPLAST Conference*, 2010.

[29] Yaron Lipman and Ingrid Daubechies. Conformal wasserstein distances: Comparing surfaces in polynomial time. *Advances in Mathematics*, 2010.

[30] Ravish Mehra, Pushkar Tripathi, Alla Sheffer, and Niloy J. Mitra. Visibility of noisy point cloud data. *Computers and Graphics*, 34(3):219–230, 2010.

[31] M. Melhi, S. S. Ipson, and W. Booth. A novel triangulation procedure for thinning hand-written text. *Pattern Recognition Letters*, 22(10):1059–1071, 2001.

[32] Xiaofeng Mi, Douglas DeCarlo, and Matthew Stone. Abstraction of 2D shapes in terms of parts. In *Proceedings of the 7th International Symposium on Non-Photorealistic Animation and Rendering*, pages 15–24. ACM, 2009.

[33] Asish Mukhopadhyay and Augustus Das. Curve reconstruction in the presence of noise. In *Proceedings of the Computer Graphics, Imaging and Visualisation Conference*, pages 177–182, 2007.

[34] Patrick Mullen, Fernando de Goes, Mathieu Desbrun, David Cohen-Steiner, and Pierre Alliez. Signing the unsigned: Robust surface reconstruction from raw pointsets. *Computer Graphics Forum, proceedings of the EUROGRAPHICS Symposium on Geometry Processing*, 29(5):1733–1741, 2010.

[35] Patrick Mullen, Pooran Memari, Fernando de Goes, and Mathieu Desbrun. Hot: Hodge-optimized triangulations. *SIGGRAPH (to appear)*, 2011.

[36] F. Preparata and M. Shamos. *Computational Geometry: An Introduction.* Springer-Verlag, 1985.

[37] Yossi Rubner, Carlo Tomasi, and Leonidas J. Guibas. The earth movers distance as a metric for image retrieval. *International Journal of Computer Vision*, 40:2000, 2000.

[38] O. Ruiz, Carlos A. Vanegas, and C. Cadavid. Ellipse-based principal component analysis for self-intersecting curve reconstruction from noisy point sets. *The Visual Computer*, 27(3):227, 2011.

[39] Yuqing Song. Boundary fitting for 2d curve reconstruction. *The Visual Computer*, 26:187–204, 2010.

[40] Cédric Villani. *Topics in Optimal Transportation.* American Mathematical Society, 2010.

[41] Jianhua Wu and Leif Kobbelt. Fast mesh decimation by multiple-choice techniques. In *Proceedings of the Vision, Modeling, and Visualization Conference*, pages 241–248, 2002.