An Optimal Transport Approach to Robust Reconstruction and Simplification of 2D Shapes
Fernando de Goes, David Cohen-Steiner, Pierre Alliez, Mathieu Desbrun

To cite this version:

HAL Id: hal-00758019
https://inria.hal.science/hal-00758019
Submitted on 27 May 2013

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L’archive ouverte pluridisciplinaire HAL, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d’enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.
An Optimal Transport Approach to Robust Reconstruction and Simplification of 2D Shapes

Fernando de Goes\textsuperscript{1} \quad David Cohen-Steiner\textsuperscript{2} \quad Pierre Alliez\textsuperscript{2} \quad Mathieu Desbrun\textsuperscript{1}

\textsuperscript{1}Caltech \quad \textsuperscript{2}INRIA Sophia Antipolis - Méditerranée

Abstract
We propose 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.

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 wide 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 \textit{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.

Instead, 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.

Reconstruction. For noise free datasets, existing reconstruction methods vary mostly based on sampling assumptions. Uniformly sampled datasets can be dealt with using image thinning [MIB01], alpha shapes [EKS83] or r-regular shapes [Att97]; for non-uniform sampling, most provably correct methods rely on Delaunay filtering [ABE98], with improvements on computational efficiency [GS01], sampling bounds [DK99, DMR99], and handling of corners and open curves [FR01, DW02].

Noisy datasets have been tackled by a variety of methods over the last ten years [Lee00, CFG\textsuperscript{*}03, MD07, MTSM10], with recent successes including the extraction of self-intersecting curves [RVC11]. Most of these reconstruction approaches first perform noise removal through clustering, thinning, or averaging, 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 [Son10] and robust statistics [FCOS05], to k-order alpha shapes [KP08], spectral methods [KSO04] and $\ell_1$-minimization [ASGCO10]—but often at the cost of a
loss of sharp features and/or a significant increase in computational complexity.

**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 [AHPMW02, GZ05], whereas coarse-to-fine methods—such as the Douglas-Peucker algorithm (see [HS94])—proceed through refinement instead. When topological guarantees are called for, a fine-to-coarse strategy is often preferred; for example, [DDS09] uses 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 [AH-PMW02] 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 [MDS09, DGDV11], whereas coarse-to-fine methods—such as the Douglas-Peucker algorithm (see [HS94])—proceed through refinement instead.

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 piece-wise 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.

## 2 Approach

We first review the basics of optimal transport before providing details on how we formulate our reconstruction task using this powerful framework.

### 2.1 Optimal Transport Formulation

Optimal transport refers to the problem of optimizing the cost of transportation and allocation of resources [Vil10] (for applications in geometric computing, see [LD10, Mém11, MMDD11]). 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; this restriction was relaxed by Kantorovich who extended the formulation to deal with transport plans between two probability measures \( \mu \) and \( \nu \). In this formulation, a transport plan is a probability measure \( \pi \) on \( \text{support}(\mu) \times \text{support}(\nu) \) whose marginals are \( \mu \) and \( \nu \). If \( \mu \) and \( \nu \) 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

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

© 2011 The Author(s)

© 2011 The Eurographics Association and Blackwell Publishing Ltd.
To reuse the example mentioned above, if each distribution is viewed as a unit amount of piled-up sand, this metric is the minimum cost of turning one pile into the other when the $L_2$ distance is used to measure the cost of transport. Note that the measures $\mu$ and $\nu$ are assumed to be probability distributions, hence both with unit total mass; throughout this paper, we adopt a trivial extension of the $W_2$ distance in $\mathbb{R}^2$ that incorporates the total amount of mass to be transported. More specifically, for any pair of measures $\mu$ and $\nu$ with bounded support in $\mathbb{R}^2$ and common total mass $M$, we define the optimal transport cost $W_2$ as

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

(1)

Our general approach to reconstruct a simplicial complex $T$ from a point set $S$ can thus be phrased as follows:

Considering $S$ as a measure $\mu$ consisting of Dirac masses, find a coarse simplicial complex $T$ such that $\mu$ is well approximated by a linear combination of uniform measures on the edges and vertices of $T$.

The main advantage of the notion of optimal transport distance in the context of reconstruction is its robustness to noise and outliers [CCSM09]. Moreover, it captures two complementary notions of approximation between $S$ and $T$. 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).

### 2.2 Transport Plan & Cost from Points to Simplices

Denote by $S = \{p_i\}$ the input point set, for which every point $p_i$ is seen as a Dirac measure $\mu_i$ centered at $p_i$ and of mass $m_i$. The point set is thus considered as a measure $\mu = \sum_\{p_i\} \mu_i$. Let us assume for now that we are given a triangulation $T$, and a point-to-simplex assignment which maps every input point $p_i$ to either an edge $e$ or a vertex $v$ of $T$—we will explain in Sec. 2.3 how this assignment is automatically determined. We regard each 0- and 1-simplex of $T$ as a uniform measure supported over the simplex itself, that is, $\nu$ is also seen as a Dirac measure, but $e$ is a uniform 1D measure defined over the edge $e$. Equipped with these measures and knowing the prescribed point-to-simplex assignment, we can now derive a transport plan $\pi$ from $S$ to vertices and edges of a triangulation $T$.

Consider an arbitrary vertex $v$ and an arbitrary edge $e$. Denote by $\mathcal{S}_v$ the set of points of $S$ assigned to the vertex $v$ and by $\mathcal{S}_e$ the set of points of $S$ assigned to the edge $e$; we assume these sets to be disjoint, with $\bigcup_{v \in T} \mathcal{S}_v \cup \bigcup_{e \in T} \mathcal{S}_e = S$.

We further call $M_v$ the total mass of $\mathcal{S}_v$ and $M_e$ the total mass of $\mathcal{S}_e$—thus, $\sum_{v \in T} M_v + \sum_{e \in T} M_e = M$. Finally, we denote by $\pi$ the plan satisfying the prescribed assignments from points to simplices, and $W_2(\pi)$ its transport cost.

**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

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

(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 transport plan into a normal and a tangential component to $e$. The normal plan is a simple orthogonal projection, and its transport cost $N$ is expressed as

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

(3)

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 $\text{card}(\mathcal{S}_e)$ segment bins, where the $i$-th bin length is $l_i = (m_i/M_e)|e|$. Each point

![Figure 2: Boundary and feature preservation. Left: Boundaries are preserved as an edge extending beyond (or falling short of) the point set would incur a large tangential component. 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 also induce alignment of edges to features.](image-url)

![Figure 3: Leaf. Example with sharp corners, boundaries, and branching points.](image-url)
The total $W_2$ cost to transport $S$ to $T$ through the transport plan $\pi$ is thus conveniently written by summing the contributions of every edge and vertex of $T$:

$$W_2(\pi) = \sqrt{\sum_{e \in T} \left[N'(e,S_e)^2 + T(e,S_e)^2\right] + \sum_{v \in T} W_2(v, S_v)^2}.$$  

(5)

### 2.3 Point-to-Simplex Assignment

Given a triangulation $T$, we still need to define an assignment of the input point set $S$ to 0- and 1-simplices of $T$ which minimizes the total cost of the transport plan $\pi$ 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 $S$ to edges and vertices of $T$ in two steps. Each point $p_i$ is first temporarily assigned to the closest edge of the simplicial complex, resulting in a partition of $S$ into a disjoint union of subsets $S_e$, where each $S_e$ 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 $S_e$ assigned to $e$, or (ii) assign instead each point of $S_e$ to its closest endpoint of $e$. Using Eqs. (2)-(4) 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 $T$ all have their assigned points (respectively, $S_v$ and $S_e$) from $S$, and the transport plan (and cost) for this assignment is thus fully determined as derived in Sec. 2.2.

Note that this assignment provides a natural characterization of the edges of $T$: an edge $e$ is called a ghost edge when $S_e = \emptyset$, 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. 1, center-right).

### 3 Algorithm

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

**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. 1 and give pseudocode in Fig. 4.

#### 3.1 Initialization

Our algorithm begins by constructing a 2D Delaunay triangulation $T_0$ of the input point set $S$. 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 a uniform measure over $e$.

$q_i$ 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 $S_e$ 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 $q_i$,

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

We can now sum up the cost $t_i$ for every point $p_i$ in $S_e$ to get the tangential component $T$ of the optimal transport cost for an entire edge:

$$T(e, S_e) = \sqrt{\sum_{p_i \in S_e} m_i \left( \frac{l_i^2}{12} + c_i^2 \right)}. \tag{4}$$

Note that Eq. (4) 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.

The pseudocode of the algorithm is given in Fig. 4.

---

**Figure 4:** Pseudocode of the algorithm.

© 2011 The Author(s)

© 2011 The Eurographics Association and Blackwell Publishing Ltd.
ing that the sample points fall within the convex hull of \( T_0 \).

With this triangulation is associated a trivial optimal plan \( \pi_0 \) for which each point is assigned to its corresponding vertex in \( T_0 \) for a total transport cost of zero.

### 3.2 Simplification

Simplification of the initial mesh \( T_0 \) is performed through half-edge collapses, each one removing one vertex and three edges from the triangulation (Fig. 5). As a collapse turns the current triangulation \( T_k \) into a new triangulation \( T_{k+1} \), it induces a local change of the transport plan \( \pi_k \) between \( S \) and \( T_k \) and, consequently, a change of the total cost \( W_2(\pi_k) \) (denoted \( \delta_k \) hereafter, with \( \delta_k = W_2(\pi_{k+1}) - W_2(\pi_k) \)).

As our goal is to minimize the increase in total cost, we apply collapses in increasing order of \( \delta \). To this end, we initially simulate all feasible collapses to determine their associated \( \delta \) and fill up a dynamic priority queue sorted in increasing order of \( \delta \). 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. 2.3, 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. 6, 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 \( \delta \) impacted by the new point-to-simplex assignment (Fig. 6, right).

**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 corre-

Figure 5: **Edge simplification.** Half-edge collapse (arrow) triggers a local reassignment of the input points (assignments: red for edges, green for vertices).

Figure 6: **Updates during collapse.** Left: collapse operator. Middle: edges with updated cost. Right: collapse operators pushed to the priority queue.

Figure 7: **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.

Suppose we want to collapse a half-edge \((x_i, x_j)\). We denote by \( P_{xy} \) the (counter-clockwise oriented) polygon formed by the one-ring of \( x_i \) and \( K_{xy} \) its kernel. We say that an edge \((a,b) \in P_{xy} \) is blocking \( x_i \) 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 \( K_{xy} \) in the direction of \((x_i, x_j)\) until it includes \( x_i \). For every blocking edge \((a,b) \in P_{xy} \), 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 \( K_{xy} \) can be elongated by removing \((a,b) \) from \( P_{xy} \). 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 \( P_{xy} \). We remove an edge \((a,b) \) from \( P_{xy} \) by flipping either the edge \((a, x_i)\) or the edge \((b, x_j)\). Note that, since \((a,b)\) has the greatest distance \( D \) within \( P_{xy} \), either \((a, x_i)\) or \((b, x_j)\) 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 \( K_{xy} \) the most. By repeating edge flips, we iteratively move \( K_{xy} \) towards \( x_j \) and this procedure is guaranteed to terminate when the modified \( P_{xy} \) no longer has blocking edges. It is worth pointing out that \( x_j \) may not be inside the final \( P_{xy} \); however, \( x_i \) is later deleted by the collapse. Fig. 7 illustrates this deterministic flipping procedure.

### 3.3 Vertex Relocation

The simplification scheme presented so far restricts the successive triangulations \( \{T_k\} \) obtained through simplification to have vertices on input points. However, noise and miss-
ing 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 [GZ05]) after each half-edge collapse (Fig. 8).

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 $W_2$ distance (defined in Eq. (2) and Eq. (3)). Remembering that the square of the normal part of the gradient of the above expression to zero. If we denote by $\lambda_i$ the barycentric coordinate of the projection $q_i$ of the input point $p_i$ onto an edge $(v, b)$, i.e., $q_i = (1 - \lambda_i)v + \lambda_ib$, then the optimal position $v^*$ of $v$ is given by:

$$v^* = \frac{\sum_{p_i \in S_v} m_i p_i + \sum_{b \in N_1(v) \cap S_b} \sum_{p_i \in S(u)} m_i (1 - \lambda_i)(p_i - \lambda_i b)}{M_v + \sum_{b \in N_1(v) \cap S_b} \sum_{p_i \in S(u)} m_i (1 - \lambda_i)^2}.$$ 

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 $\mathcal{S}_{k+1}$.

### 3.4 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. 2.3, 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. 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

$$r_e = \frac{M_e[|e|^2]}{N(e, S_e)^2 + T(e, S_e)^2}.$$ 

In our experiments, a threshold on relevance was only applied to examples with large amount of outliers (Figs. 1, 10, and 17).

### 3.5 Implementation details

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

**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.

**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 [WK02] 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 $\delta$. 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. 1).

**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.

**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 discards 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. 9 is the only figure where this option was activated).

### 4 Results

We implemented our reconstruction and simplification algorithm using the CGAL library [CGA]. 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.

On closed and noise-free curves, our approach performs...
Figure 10: 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.

Figure 9: Tolerance Control. A normal and tangential tolerance error (tolerance in gray, reconstruction in green) are imposed during simplification on the shark model.

Table 1: Timing. 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.

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

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 [GS01] fail to properly reconstruct a point set as soon as noise is added (Fig. 11); Poisson-like implicit approaches [KBH06] have also been reported to fail in this case unless reliable normals are provided [ACSTD07]. In contrast, our transport-based method is robust to large amounts of noise and outliers as illustrated in Fig. 10 where more outliers than initial points still leads to a good reconstruction—and sharp features are still captured (see also Fig. 12) without the typical blunting that most previous work (based on local denoising) would create. Fig. 13 depicts how the reconstruction gracefully degrades when the point density decreases. Our method also handles complex shapes with branching, intersections, and open curves without additional parameters to tweak (see Fig. 3).

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 consid-

Figure 11: Bird. Top: For closed, noise-free point sets (left), combinatorial methods such as crust [GS01] (or other crust variants) correctly connects all input points (middle), while we obtain a simplified reconstruction (right). Bottom: With added synthetic noise (0.5% of bounding box), combinatorial approaches fail to recover the shape, while our method (right) returns a result nearly identical to the noise-free case.
Figure 12: 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 13: Light house. 20-vertex reconstructions from input points with decreasing sampling density (from left to right: 2K, 400, and 200 points resp.).

Figure 14: 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 15: 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 18: Gallery. Plate of shapes reconstructed through our optimal transport driven approach with input points originating from noiseless dataset (cartoon face contour), scanned drawings (screwdriver, tablecloth), grey-level images (kanji characters, flames), and image gradients computed by finite difference (all others).

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; however, new decimation and transport cost evaluation strategies will be needed to offer both robustness and efficiency.

Acknowledgments. We thank Patrick Mullen for discussions along the way, and Xiaofeng Mi, Doug DeCarlo, and Ravish Mehra for providing data. This work was funded by the European Research Council (ERC Starting Grant “Robust Geometry Processing”, Grant agreement 257474), and by NSF grants (CCF-0811373, CMMI-0757106, and CCF-1011944).

References


[ACST07] ALLIEZ P., COHEN-STIEER D., TONG Y., DESBRUN M.: Voronoi-based variational reconstruction of unori-


[Son10] Song Y.: Boundary fitting for 2d curve reconstruction. The Visual Computer 26 (2010), 187–204. 1
