Mass-driven Topology-aware Curve Skeleton Extraction from Incomplete Point Clouds

Hongxing Qin[1] Jia Han[1] Ning Li[1] Hui Huang[2*] Baoquan Chen[3]
[1]Chongqing University of Posts and Telecommunications, Key Laboratory of Data Engineering and Visual Computing, College of Computer Science and Technology, China
[2]Shenzhen University, College of Computer Science and Software Engineering, China
[3]Peking University, Center on Frontiers of Computing Studies, China

We introduce a mass-driven curve skeleton as a curve skeleton representation for 3D point cloud data. The mass-driven curve skeleton presents geometric properties and mass distribution of a curve skeleton simultaneously. The computation of the mass-driven curve skeleton is formulated as a minimization of Wasserstein distance, with an entropic regularization term, between mass distributions of point clouds and curve skeletons. Assuming that the mass of one sampling point should be transported to a line-like structure, a topology-aware rough curve skeleton is extracted via the optimal transport plan. A Dirichlet energy regularization term is then used to obtain a smooth curve skeleton via geometric optimization. Given that rough curve skeleton extraction does not depend on complete point clouds, our algorithm can be directly applied to curve skeleton extraction from incomplete point clouds. We demonstrate that a mass-driven curve skeleton can be directly applied to an unoriented raw point scan with significant noise, outliers and large areas of missing data. In comparison with state-of-the-art methods on curve skeleton extraction, the performance of the proposed mass-driven curve skeleton is more robust in terms of extracting a correct topology.

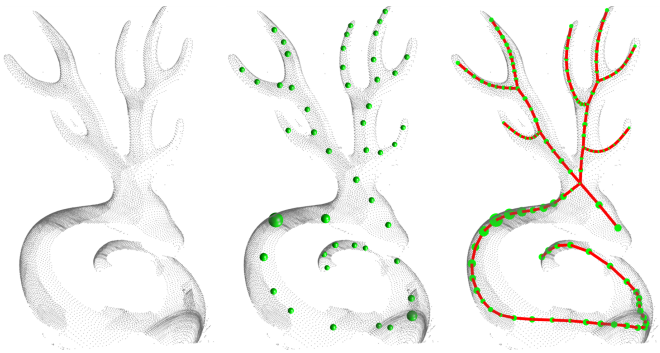
***Index Terms*—Curve skeleton, incomplete point clouds, optimal mass transport, Wasserstein distance.**

I. Introduction

The skeleton of a shape provides an intuitive and effective abstraction that facilitates shape understanding and manipulation. The skeletons have been extensively used in various applications, such as shape registration, model segmentation, animation, morphing, shape recognition, and shape retrieval. Since the medial axis was introduced by Blum [1], it has become the basis for many shape descriptors [2]. The medial axis of a 3D model is generally a non-manifold structure that is composed of 2D surface sheets, and is difficult to store and manipulate. The curve skeleton of a 3D model is a line-like representation. In comparison with a medial axis, a curve skeleton is more extensively adopted given the simplicity of topology and ease of manipulation in the computer graphics community [3]–[5]. In this work, we focus on curve skeleton extraction from incomplete point clouds.

Many algorithms can be used for curve skeleton extraction [3], [6], but most of these algorithm are designed to deal with shapes specified by closed polygonal meshes. In contrast to meshes, no explicit topology connection exists between unorganized point samples, which makes curve skeleton extraction from point clouds extra challenging. Furthermore, raw scans typically contain large gaping holes, severe undersampling, heavy noise, and outliers. Two key problems, namely, *robustness* and the *accuracy*, must be carefully addressed.

Another thread of algorithms [4], [5] first extract dense points as a branch of rough curve skeleton and then construct connections between different branches by estimating the bridging points at joints. Various branches of the skeleton may require to be extracted using different parameters, thereby making the process tedious. When large regions of missing data are available in the input point clouds, the difficulty rapidly increases in constructing a correct topology graph.



(a) Raw scan (b) Mass points (c) Mass skeleton

Fig. 1: Given an unorganized, unoriented, and highly incomplete raw scan corrupted with noise and outliers, we are able to extract a complete and quality curve skeleton with respect to mass distribution.

In this paper, we introduce a mass-driven curve skeleton (MdCS) that simultaneously presents geometric properties and mass distribution of a curve skeleton as a new type of curve skeleton representation for 3D point cloud data. MdCS is defined as a solution minimizing Wasserstein distance with an entropic regularization term between two probability measures, namely, mass distribution of an original point cloud and a MdCS.

To generate skeletons with an accutrate topology from incomplete point clouds, we first extract a topology-aware rough curve skeleton with sparse points (Fig. 1[b]) rather than dense points, similar to that in existing works [4], [5]. By interpolating additional points in the rough curve skeleton, a

*Corresponding author: Hui Huang (hhzhiyan@gmail.com).

smooth MdCS (Fig. 1[c]) is constructed via an optimization procedure. Numerical experiments show that our algorithm can produce satisfactory curve skeletons, even when input point clouds are heavily corrupted. In comparison with ℓ_1-medial skeleton [5], our algorithm is more robust because it does not depend on suitable neighborhood sizes, which are essential in ℓ_1-medial skeleton [5]. Our algorithm naturally provides a multi-scale curve skeleton in terms of the mass of each point sampled in the curve skeleton.

The main theoretic and algorithmic contributions are presented as follows:

- Curve skeleton points are represented as a finite set of centers of power diagram cells. Curve skeleton extraction is formulated as an optimal mass transport problem.
- A MdCS as a new type of skeleton not only presents a geometry position as other methods, but also provides a mass distribution, of a curve skeleton.
- In comparison with state-of-the-art methods, the proposed algorithm performs robustly on unorganized, unoriented, and highly incomplete point cloud data.

II. Related work

A. Curve skeleton extraction from point clouds

The literature on skeletal shape extraction is vast. Here, we only briefly review works on curve skeleton extraction from point clouds. For the rest, we refer readers to two comprehensive surveys [3], [6].

Ogniewicz and Ilg [7] proposed a Voronoi skeleton method for point clouds via a Voronoi diagram of boundary points. On the basis of a deformable model evolution, Sharf et al. [8] tracked the front of a smooth blob from the inside of a point cloud to generate a curve skeleton. Cao et al. [9] presented an algorithm for curve skeleton extraction via Laplacian-based contraction (LBC) by extending the mean curvature flow based skeleton extraction method [10] to point clouds. However, these algorithms require the input point cloud to be sufficiently clean and dense to generate a correct curve skeleton. Moreover, these algorithms may be ineffective for curve skeleton extraction from raw scan point clouds that are severely under-sampled, with heavy noise and outliers.

Tagliasacchi et al. [4] proposed a rotational symmetry axis (ROSA) method to extract curve skeletons for point clouds with considerable missing data. This method relies on a cylindrical shape prior and accurate point normals. Livny et al. [11] proposed an algorithm for extracting a skeletal structure from tree data. Li et al. [12] developed a deformable model called arterial snakes for curve abstraction, which is targeted at input models that can be represented well by 1D curves. Works [13]–[15] constructed curve skeletons by extracting a Reeb graph of point clouds. Song et al. [16] developed a distance field guided ℓ_1-median method to extract curve skeletons from points. Zhang et al. [17] proposed a skeletal representation of a tree on internodes. In Table I, we summarize related curve skeleton extraction methods of point clouds and their performance level of robustness to various artifacts.

Works [5], [18] and [19] are studies that are most related to our work. Huang et al. [5] proposed an ℓ_1 projection operator to extract curve skeletons from point clouds. ℓ_1 projection operator can be directly applied to an unoriented raw point scan. Nonetheless, accurate parameters are difficult to set during an iterative procedure to obtain satisfactory results. On the basis of a topology structure prior, Lu et al. [18] constructed the curve skeleton of a shape using centroidal Voronoi tessellation. Solomon et al. [19] constructed a curve skeleton of a shape using Wasserstein propagation. Mérigot et al. [20] proposed an algorithm for the optimal transport between a simplex soup and a point cloud. In comparison with [5], we represent a curve skeleton as the Wasserstein barycenter of the probability measures of point clouds, rather than ℓ_1-median, and improve the robustness of the curve skeleton extraction method. In comparison with [18] and [19], our algorithm does not require a topology structure prior but directly estimates a topology graph of point clouds on the optimal transport plan. We represent point clouds and their curve skeleton as two probability measures to calculate Wasserstein distance, rather than using barycentric coordinates [21], as probability measures similar to those in [19]. De Goes et al. [22] and Digne et al. [23] proposed optimal transport approaches for the robust reconstruction of 2D and 3D shapes, respectively. In comparison with their approaches, our method focuses on extracting a topology structure rather than preserving a geometric detail. Furthermore, our approach is based on a simpler transport plan from 0-simplices to 0-simplices rather

TABLE I: Categorization of curve skeleton extraction of point clouds in terms of the ability to handle point cloud artifacts and input requirements. Here, the symbol ○ indicates that the method is moderately robust to a particular artifact, and ● indicates that the method is robust. We use ✓ to indicate that input requirements are necessary. Non-uniform sampling denotes that the sampling density spatially varies. Noise indicates that sampling points are randomly distributed near the surface. Missing data denote that the sampling density is 0 in the region. Normals indicate that the normal of the point is the direction perpendicular to the tangent space of the point. Tubular represents an appropriate shape prior, and the shapes of interest must be covered by a generally tubular region, except at their joints.

Method	Point Cloud Artifacts			Input Requirements	
	Non-uniform sampling	Noise	Missing data	Normals	Tubular
Voronoi [7]	○	○	○		
LBC [9]	○	○	○		
ℓ_1 [5]	○	○	●		
Reeb Graph [14]	●	○	○		
ROSA [4]	○	○	●	✓	✓
Snake [11]	○	○	●	✓	✓
Tree skel [12]	○	●	●		✓

than {0,1,2}-simplices used in [22] and [23].

B. Optimal mass transport and curve skeleton extraction

Recently, optimal mass transport has been extensively applied to geometry processing [19], [22], [24], [25]. For completeness, we briefly review the Wasserstein distance in this subsection. For an arbitrary space Ω, we use $d : \Omega \times \Omega \to R_+$ to denote the distance metric, that is, $d(x, y)$ is the shortest distance from x to y along Ω. We use $P(\Omega)$ to indicate the set of probability measures on Ω, and $P(\Omega \times \Omega)$ to denote that on the product space $\Omega \times \Omega$,

$$P(\Omega) = \{\mu \mid \mu(\varnothing) = 0,\ \mu(\Omega) = 1,\ 0 \le \mu(U) \le 1(U \subseteq \Omega)\},$$

where $\mu(U) = \int_{U\in\Omega} \rho(x)dx$ is a probability measure on Ω, and $\rho(x) \ge 0$ is a probability density function on Ω.

Given probability measures μ and ν in $P(\Omega)$ and $p \ge 1$, the p-Wasserstein distance [26] between μ and ν is defined as:

$$W_p(\mu, \nu) = \left(\inf_{\pi \in \prod(\mu,\nu)} \int_{\Omega\times\Omega} d(x, y)^p d\pi(x, y) \right)^{1/p},$$

where $d(x, y)$ is the cost of transporting one unit of mass from x to y. In the present work, we represent $d(x, y)$ as the Euclidean distance between the points x and y, and $\prod(\mu, \nu)$ is the subset of $P(\Omega \times \Omega)$ that satisfies mass conservation laws.

$$\prod(\mu, \nu) = \{\pi \in P(\Omega, \Omega) \mid \pi(\cdot, \Omega) = \mu,\ \pi(\Omega, \cdot) = \nu\},$$

where π is a transportation plan that describes the amount of mass $\pi(x, y)$ to be placed from μ at x towards y to create ν in aggregation. Wasserstein distance, therefore, describes the minimum cost of transporting the source μ to the target ν.

The 2-Wasserstein distance, which uses the L2 norm, is most common and is frequently referred to as the earth mover's distance. Aurenhammer et al. [27] verified that capacity constrained partitions that minimize the 2-Wasserstein distance for a given point set are power diagrams. Furthermore, De Goes et al. [24] and Qin et al. [25] exploited the nature of the 2-Wasserstein distance to obtain the center of power diagram cells for blue noise sampling. In our work, a power diagram partition of given point cloud data is developed by optimizing the 2-Wasserstein distance, and curve skeleton points are represented as the center of power diagrams cells. In power crust algorithm [28], [29], Amenta et al. considered the points on the medial axis as the center of power diagram cells and used the boundary of the power diagram cells to approximate a surface. In comparison with [28] and [29], our approach for curve skeleton extraction is an approximate inverse procedure of the power crust algorithm.

The principle of attributing points with a mass is well known in many skeletonization methods, such as in [30]–[33]. In these methods, the mass of a surface is defined as the product of the surface area and mass density. This mass is considered uniform. Therefore, the mass of a sampling point is proportional to the area element represented by the point. In [30]–[33], the sampling points are uniformly distributed and the mass of the points is the same. The skeleton detection and the mass of skeleton points is modeled via an advection principle that collapses mass from the surface to the surface skeleton, then to the curve skeleton, and finally to the latter's center. In comparison with the collapse procedure in these methods, the mass of a sampling point is estimated in terms of the local sampling density in our approach, and the curve skeleton points are defined as the center of the power diagram cell. The mass in the power diagram cell is directly transported to the center. The mass in [30]–[33] is accumulated step by step via the collapse procedure. In these methods, the topology structure of curve skeleton is preserved during the collapse procedure. In our approach, the topology structure of a curve skeleton is extracted from the transport plan Π. Locally defined principal curves [34] are also modeled on mass distribution. The principal curves are defined as the local maximum of the probability density function in the orthogonal space defined by the local Hessian matrix of the probability density function. A fixed point iteration algorithm, which is analogous to mean-shift algorithm, is provided for the principal curves extraction.

III. Overview

The input to our algorithm is an unorganized set of points $Y = \{y_j\}_{j\in J} \subset R^3$, which are unoriented, thereby containing large gaping holes and severely corrupted. The output is an MdCS, which is a $1D$ curve skeleton $X = \{x_i\}_{i\in I} \subset R^3$ with mass distribution $\{\rho_i\}$ and topology connection. Here, X represents a 1D local center of the shape underlying the input Y, and ρ_i represents the mass that is transformed from Y to x_i. To reduce the impact of noise and missing data on topology extraction and improve the robustness of our algorithm, we initially extract a rough curve skeleton with correct topology, rather than a smooth curve skeleton directly. The pseudocode is demonstrated in Algorithm 1 and the main algorithmic phases are presented as follows:

a) ***Topology extraction:*** Given that a curve skeleton of a shape is a 1D structure, a rough MdCS of the shape is extracted via the relaxed optimal transport problem between two discrete probability measures. Additional details can be found in Section IV-B.

b) ***Topology checking:*** The topology structure of curve skeletons are checked via the symmetry detection of the optimal transport plan, and the incorrect topology structure is alleviated by computing a new optimal transport plan on a large-scale parameter. Further details can be found in Section IV-C.

c) ***Geometry optimization:*** A smooth curve skeleton is generated by minimizing the Wasserstein distance with a Dirichlet energy regularization. Additional details can be found in Section IV-D.

IV. Mass-driven curve skeleton

A. Defining a mass-driven curve skeleton

An unorganized set of points $Y = \{y_j\}_{j\in J} \subset R^3$ represent a discrete probability measure ν in R^3.

$$\nu = \sum_{j=1}^{n} \varrho_j \delta_{y_j} \quad s.t. \sum_{j=1}^{n} \varrho_j = 1,\ \delta_{y_j} = \begin{cases} 1 & y_j \in Y, \\ 0 & \text{others}, \end{cases}$$

where n is the number of samples, ϱ_j is the probability at y_j, and J indexes the set of input points Y.

An 1D curve skeleton point set $X = \{x_i\}_{i\in I} \subset R^3$ represents another discrete probability measure μ.

$$\mu = \sum_{i=1}^{m} \rho_i \delta_{x_i} \quad s.t. \sum_{i=1}^{m} \rho_i = 1, \, \delta_{x_i} = \begin{cases} 1 & x_i \in X, \\ 0 & \text{others}, \end{cases}$$

where m is the number of sampling points in the curve skeleton, ρ_i is the mass or probability at x_i, and I indexes the set of sampling points of the curve skeleton.

We investigate the following definition for an MdCS that leads to an optimal discrete probability measure μ:

$$\arg\min_{\mu} (1-\lambda) W_2^2(\mu, \nu) + \lambda R(X), \tag{1}$$

where $W_2^2(\mu, \nu)$ is the Wasserstein distance between the probability measures μ and ν. the second term $R(X)$ regularizes the local point distribution X, and $\lambda \in [0, 1]$ is the weight parameter to balance the Wasserstein distance and the regularization term.

The first term in Eq. (1) denotes that the method aims to find a μ that captures ν well. In terms of the theory that capacity constrained partitions that minimize the 2-Wasserstein distance for a given point set are power diagrams [27], the solution of minimizing the first term provides power diagram partitions of original point clouds, and x_i is the center of the corresponding power cell. However, these points x_i may represent a surface similar to that in [25], rather than a curve skeleton. The object here is to compute a skeletal point cloud that provides a $1D$ representation of the input geometry. In particular, we are seeking an optimal probability measure μ with a line like structure to represent a shape defined by ν . The second term, therefore, is added to regularizes the curve fitting and avoids unwanted oscillations.

In comparison with the power crust [29] that uses the power diagram to approximate the medial axis, the main idea of our algorithm is to use a finite set center of power cells to approximate the curve skeleton. For a tubular model, the medial axis degenerates to the curve skeleton. Therefore, the quality of point set Y, which represents the input shape F in our algorithm, must comply with *r-sample* [29] from the theory view. Y is an *r-sample* from F when the distance from any point $y \in F$ to the nearest sample y_i is at most $rLFS(y)$, and $LFS(y)$ is the minimum Euclidean distance from point y to any point of the medial axis. In practice, we aim to implement the algorithm to provide favorable results even when this assumption is unsatisfied. In comparison with ℓ_1-medial skeleton [5], our definition provides a global optimal transport plan to represent the mapping between Y and X, rather than using a local Gaussian decay weighting function. Furthermore, the optimal transport plan also provides the cue to extract topology structures of a curve skeleton. The details of the optimal transport plan will be introduced in the next subsection.

B. Topology extraction

We use the fact that the curve skeleton of a shape is a $1D$ structure, and the optimal transport plan between probability measures μ and ν is determined to extract the topology graph of a shape via an iterative contraction.

For topology extraction, we must only extract a rough MdCS with correct connection, and an accurate curve fitting is unnecessary. Therefore, the first term in Eq. (1) is only used to extract skeleton points, which are modeled as the minimum of the Wasserstein distance. That is,

$$\mu = \arg\min_{\mu} W_2^2(\mu, \nu) = \arg\min_{\mu} \inf_{\pi} \int_{X\times Y} d(x_i, y_j)^2 d\pi(x_i, y_j) \\ s.t. \sum_{j=1}^{n} \pi(x_i, y_j) = \rho_i, \sum_{i=1}^{m} \pi(x_i, y_j) = \varrho_j. \tag{2}$$

Let $V^{\omega}(x_i)$ be the power diagram [35] region of x_i.

$$V^{\omega}(x_i) = \{x \in R^3 \mid ||x - x_i|| - \omega_i \leq ||x - x_k|| - \omega_k, \forall k \neq i\}.$$

The optimal solution π of Eq. (2) establishes a power diagram partition of point set $X = \{x_i\}$; that is:

$$\pi(x_i, y_j) = \begin{cases} \varrho_j & y_j \in V^{\omega}(x_i), \\ 0 & \text{others}. \end{cases} \tag{3}$$

However, a power diagram partition is a hard partition as well as a Voronoi diagram partition, and is sensitive to heavy noise and outlier of point clouds [7]. We, therefore, introduce an entropic regularization term similar to that in [36] to relax the transport plan and break the condition of Eq. (3).

The regularized Wasserstein distance is defined as:

$$W_2(\mu, \nu) = \left(\inf_{\pi \in \prod(\mu,\nu)} \int_{\Omega\times\Omega} d(x, y)^2 d\pi(x, y) - \epsilon H(\pi) \right)^{1/2}, \tag{4}$$

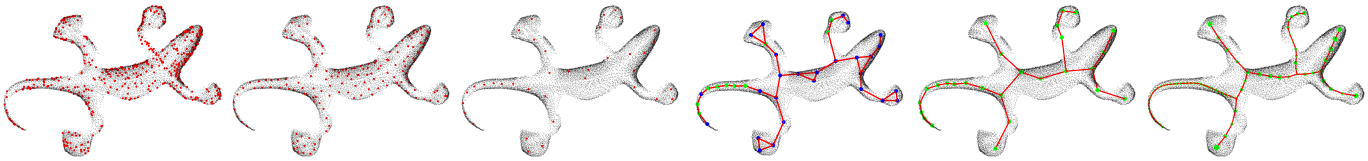
where ϵ is a positive regularization parameter, and

$$H(\pi) = -\int_{\Omega\times\Omega} \pi(x, y) ln \pi(x, y) dx dy,$$

is the entropy of π. This term expands the transport range, rather than the nearest transport in Eq. (2). The parameter ϵ plays a role in controlling the overlap range of Y_i and Y_k, where $Y_i = \{y_j | \pi(x_i, y_j) > 0, y_j \in Y\}$, as illustrated in Fig. 2. Given an initial μ and ν, an updated set of points $X = \{x_i\}_{i\in I}$ and an optimal transport plan Π can be obtained from Eq. 4 by fixing $\rho_i (i \in I)$.

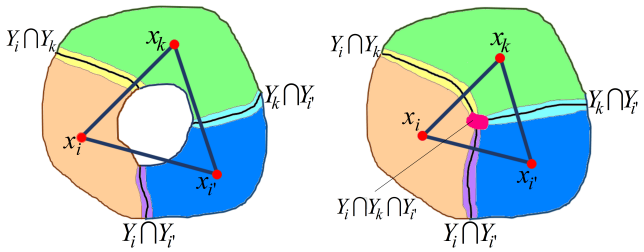
Subsequently, we initially describe the methods that we use in identifying and labeling points that belong to a rough curve skeleton. Then, we explain the merging of points that do not belong to a rough curve skeleton.

Considering that a curve skeleton is locally a line-like structure, we assume that the most mass of a sampled point y_j must be transported to a line-like structure when the parameter ϵ is sufficiently small. To label a curve skeleton point from the candidates, we initially identify if the mass of a point y_j is transported to a line structure. For every point y_j, we find three curve skeleton points $x_{i_1}, x_{i_2}, x_{i_3}$, to which the



(a) Initiation (b) $\epsilon = 0.001$ (c) $\epsilon = 0.001$ (d) $\epsilon = 0.001$ (e) $\epsilon = 0.0012$ (f) MdCS

Fig. 3: Overview of mass skeleton extraction. Given a raw scan, we randomly sample a subset (a) as shown in red dots. These points are iteratively projected and merged onto a skeletal point cloud (b-c). Topology checking is applied to find the wrong curve skeleton points (d). The points that do not satisfy the symmetry condition are shown in blue dots. The points that satisfy the symmetry condition are shown in green dots. The cycles, which are formed by blue or green dots, need to be merged furthermore. The parameter ϵ is updated, and new iterations are applied until no points need to be merged (e). The final MdCS is obtained in (f) via geometry optimization.



(a) Correct topology graph (b) Incorrect topology graph

Fig. 2: Illustration of the merging procedure. (a) When the mass of one point in the original point clouds is transported to two skeleton points at most, the mass of the point is transported to a line-like structure, and a correct topology graph is extracted. (b) When the mass of one point in the original point clouds is transported to three skeleton points at least, and the three points do not position in a line, the mass of the point is not transported to a line-like structure, and an incorrect topology graph is formed. Two of the skeleton points must be merged.

transported mass $\pi(x_{i_1}, y_j)$, $\pi(x_{i_2}, y_j)$, and $\pi(x_{i_1}, y_j)$ from y_j are the largest among $\pi(\cdot, y_j)$. The point y_j is labeled as a major point of the point x_{i_1} with the largest $\pi(x_{i_1}, y_j)$ among $\pi(\cdot, y_j)$. If the sum of the two largest $\pi(x_{i_1}, y_j)$ and $\pi(x_{i_2}, y_j)$ is larger than $0.99\varrho_j$, then we assume that the mass of the point y_j is only transported to two points and set $m(x_{i_1}, x_{i_2}) = 1$. Therefore, the mass of y_j is transported to a line-like structure. Otherwise, we assume that the mass of the point y_j is transported to three points at least, and we set $m(x_{i_1}, x_{i_2}) = m(x_{i_1}, x_{i_3}) = 1$. Then, we check whether the three points $x_{i_1}, x_{i_2}, x_{i_3}$ lie in a line-like structure. If $cos(\angle(\overrightarrow{x_{i_1}x_{i_2}}, \overrightarrow{x_{i_1}x_{i_3}})) \leq -0.9$, then the mass of the point y_j is considered transported to a line-like structure. If at least 90% of the total major points of x_i exists, the mass of which is transported to a line-like structure, then we label the point x_i as a curve skeleton point.

For convenience, a point x_i denotes the point in this paragraph, which is not labeled as a curve skeleton point. We will apply a merging procedure to gradually eliminate them. The merging procedure includes two steps. The first step is to select point pairs, which will be merged. We first select a point x_i with minimum mass in points $\{x_i\}$, which are not yet selected. Then, a point pair (x_i, x_j) is selected if x_j is the nearest neighbor of x_i, and a connection is observed between x_i and x_j. Point pair selection will be performed gradually until no point pair is found. The second step is to merge every point pair (x_i, x_j) as x_i and delete the point x_j. The coordinate of x_i is updated as $\frac{x_i\rho_i + x_j\rho_j}{\rho_i + \rho_j}$, and the probability density of x_i is updated as $\rho_i + \rho_j$.

After every merging procedure, we update the adjacent matrix $\mathbf{M}$. An existing topology connection must be preserved, that is $m(x_i, x_k) = 1$ if $m(x_j, x_k) = 1$. Furthermore, the original topology connection is based on the transport plan of only one sampling point. The topology connection can be corrupted by heavy noise. To reduce the impact of noise, we introduce a constrain condition to check the updated element $m(x_i, x_k)$. For topology points x_i and x_k, if $\sum_{y_j \in Y}(\pi(x_i, y_j) + \pi(x_k, y_j)) < 0.1(\rho_i + \rho_k)$, it denotes that the mass of very few sampling points is transported to the points x_i and x_k simultaneously, and we set $m(x_i, x_k) = 0$.

C. Topology checking

While Wasserstein distance is robust to noise and outliers, an incorrect topology connection may be obtained if the given point cloud contains large gaping holes (Fig.3[e]) or sudden-change-scale structures (Fig.14[b]). To alleviate this problem, we propose a symmetry check to correct the extracted topology structure.

We adopt the classical principal component analysis (PCA) to detect three directions, which are used to check the symmetry of the transport plan $\pi(x_i, \cdot)$ at every point x_i. At each point x_i, we compute the eigenvectors of a 3×3 covariance matrix.

$$C_i = \sum_{y_i \in Y \wedge \{y_k | \pi(x_i, y_k) > 0.01\varrho_k\}} (x_i - y_j)^T (x_i - y_j),$$

where the corresponding eigenvectors $\{\mathbf{v}_i^0, \mathbf{v}_i^1, \mathbf{v}_i^2\}$ form an

Algorithm 1 Extraction of a curve skeleton

Input: Unorganized points set $\mathbf{Y} = \{y_j\}_{j\in J} \subset R_3$; parameters m, ϵ, λ, θ and τ;
Output: Curve skeleton points set $\mathbf{X} = \{x_i\}_{i\in I} \subset R^3$; Discrete probability measure μ; Adjacent matrix $\mathbf{M}$ of curve skeleton points;

function MDCS(Y,m, ϵ, λ, θ and τ)
  ν = $\{\nu_j|\nu_j$ is the local sampling density at the point $y_j\}$;
  $\mathbf{X}$ = $\{x_i|x_i \in \mathbf{Y}, x_i \neq x_j(i \neq j), i \in \{1, 2, ..., m\}\ \}$;
  μ =$\{\mu_i = 1/m|i \in \{1, 2, ..., m\}\}$;
  $\mathbf{M}$ = $\{m_{ij} = 0|i, j \in \{1, 2, ..., m\}\}$;
  $\mathbf{L}$ = $\{l_i = 0|i \in \{1, 2, ..., m\}\ \}$;
  while $\epsilon < 0.002$ **do**
    do
      $\{\mathbf{X}, \mathbf{\Pi}\}$= Optimize($Y, \epsilon, \mathbf{X}, \mathbf{\Pi}, \nu, \mu$) ;
      Update $\mathbf{M}$ in terms of $\mathbf{\Pi}$;
      $l_i \leftarrow 1$ if x_i being labeled as a curve skeleton point;
      Find (x_i, x_j) satisfying the merging condition;
      Update $\mathbf{X}, \mathbf{\Pi}, \mu$ in terms of merging strategy;
    while No (x_i, x_j) being merged
    Find incorrect cycle structures $\mathbf{C}$;
    if size(C)=0 **then**
      `break`;
      $l_i \leftarrow 0$ if $x_i \in \mathbf{C}$;
    else
      $\epsilon \leftarrow \epsilon + \Delta\epsilon$
    end if
  end while
  Insert new points x_k in $\mathbf{X}$ in terms of Eq. 7;
  $\{\mathbf{X}, \mathbf{\Pi}\}$= Optimize($Y, \epsilon, \mathbf{X}, \mathbf{\Pi}, \nu, \mu$);
  return $\mathbf{X}, \mathbf{M}, \mu$;
end function

Algorithm 2 Extremize the variational problem of Eq. 1

Input: $Y, \epsilon, \mathbf{X}, \mathbf{\Pi}, \nu, \mu$;
Output: $\mathbf{X}$, $\mathbf{\Pi}$;

function OPTIMIZE($Y, \epsilon, \mathbf{X}, \mathbf{\Pi}, \nu, \mu$)
  Compute the distance matrix $\mathbf{D}$ between $\mathbf{X}$ and $\mathbf{Y}$;
  $E_0 \leftarrow (1-\lambda)W_p^p(\mu, \nu) + \lambda R(X)$;
  do
    Compute $\mathbf{\Pi}$ via iterative Bregman projection [37]
    $\mathbf{X} \leftarrow (1-\theta)\mathbf{X} - \theta\mathbf{B}/\mathbf{A}$;
    $E_1 \leftarrow (1-\lambda)W_p^p(\mu, \nu) + \lambda R(X)$;
    $dE \leftarrow E_1 - E_0$;
    $E_0 \leftarrow E_1$;
  while dE is convergent
end function

orthogonal frame. Then, we define the value:

$$\begin{aligned} m_i^{\alpha\beta+} &= \sum_{y_j \in Y_i} \varrho_j \quad (x_i - y_j)^T \cdot v_i^\gamma > 0, \\ m_i^{\alpha\beta-} &= \sum_{y_j \in Y_i} \varrho_j \quad (x_i - y_j)^T \cdot v_i^\gamma < 0, \\ s.t.\ \alpha, \beta, \gamma &= 0,\ 1,\ 2 \quad \text{and} \quad \alpha \neq \beta \neq \gamma, \end{aligned} \tag{5}$$

as the transported mass to x_i from two opposite directions.

To check the symmetry of the transport plan $\pi(x_i, \cdot)$ at x_i, we define a testing value as:

$$m_i^d = \max_{\alpha\beta} \|m_i^{\alpha\beta+} - m_i^{\alpha\beta-}\|/(m_i^{\alpha\beta+} + m_i^{\alpha\beta-}), \tag{6}$$

which measures the maximal difference of the transported masses to x_i from two opposite directions. If m_i^d is larger than a threshold τ, x_i does not lie in the center of a local structure. This condition can occur when large data are missed in the local structure (Fig.3[e]) or a large scale structure is present around x_i (Fig.14[b]). We, therefore, label these topology points, such as x_i, as candidates, which are represented as $\{x_c^1, x_c^2, ...\}$, and label other topology points as non-candidates.

Furthermore, we obtain several connected subgraphs G containing the candidates x_c^i by traversing the topology graph from x_c^i. In this process, the non-candidates are treated as terminal points. If more than one missing area or a large scale structure exists, an incorrect cycle structure $\mathbf{C}$ may exist in G, as depicted in Fig. 3[e] and 14[b]. To correct an incorrect MdCS, we label the points that belong to C as non-skeleton points and update the parameter $\epsilon = \epsilon + \Delta\epsilon$. Then, a new merging procedure is applied until there isn't non-skeleton point. In our algorithm, a small overlap range of Y_i and Y_j is necessary. We set $0.001 \leq \epsilon \leq 0.002$ and $\triangle\epsilon = 0.0002$.

D. Geometry optimization

To obtain a smooth MdCS, we insert additional skeleton points in the rough MdCS. Then, a geometry optimization is used to determine the final positions of points X using Eq. (1).

To preserve the smallest branch contained in input point clouds, we find the shortest edge $e_s(x_i, x_k)$ in the rough MdCS, where x_i and x_k are the vertexes of the edge e_s. For every edge $e \in E$ in the rough MdCS $G(X, E)$, if the length of the edge e is longer than the length e_s, we insert l isometric points. The number l of inserted points is set in terms of the condition, that is,

$$\frac{|e|}{l+1} \leq |e_s| < \frac{|e|}{l}, \tag{7}$$

where $|e|$ is the length of the edge e. Then, we apply a linear interpolation to set the probability measure of inserted points and a normalization procedure to update the probability measures of all skeleton points X, including original and inserted skeleton points.

If the set of points X contains additional points, the solution of Eq. (2) may lead to unwanted oscillations. To avoid unwanted oscillations, a regularization term $R(X)$ is used as in Eq. (1). Similar to the works in [38] and [18], we use Dirichlet energy (a squared Laplacian) to define the regularization term

$R(X)$, which minimizes the coordinate's variations on each one-ring neighborhood.

$$R(X) = \lambda \sum_{i=1}^{n} \|x_i - \frac{1}{v(x_i)} \sum_{x_k \in N(x_i)} x_k\|^2, \tag{8}$$

where $N(x_i)$ is the one-ring neighbors of x_i, $v(x_i)$ is the number of points $N(x_i)$, and $\lambda \geq 0$ controls the influence of the regularization term.

E. Numerical optimization

In this subsection, we discuss the computation of the solutions of Eqs. (2) and (1). The pseudocode is presented in Algorithm 2. When $\lambda = 0$, Eq. (1) degenerates to Eq. (2). We, therefore, only consider the numerical optimization of Eq. (1).

The variational problem of Eq. (1) is a non-convex optimization problem. We apply an iterative algorithm to extremize the variational problem by repeatedly performing a minimization step over X followed by a projection step over transport plans π. We initiate curve skeleton points $X = \{x_1, x_2, ..., x_m\}$ from Y via a random sampling and set $\rho_i = 1/m$, $i = 1, 2, ..., m$. By fixing a set of points X, we compute a relaxed transport plan using Eq. (4), which is discretized as:

$$\begin{aligned} W_p^p(\mu, \nu) = \arg\min_{\Pi} & < \mathbf{D}, \mathbf{\Pi} > -\epsilon H(\mathbf{\Pi}), \\ & s.t.\ \mathbf{\Pi 1} = \rho,\ \mathbf{\Pi}^T \mathbf{1} = \varrho, \end{aligned} \tag{9}$$

where $\mathbf{D}, \mathbf{\Pi} \in \mathbf{R}_+^{m \times n}$, $\mathbf{D}$ is the distance matrix, and $D_{ij} = d(x_i, y_j)^p$. $\mathbf{\Pi}$ is the transport plan matrix, and $\Pi(i, j) = \pi(x_i, y_j)$ represents the mass transported from y_j to x_i. Here, ρ and ϱ are the probability density vectors, $\rho = (\rho_1, \rho_2, ..., \rho_m)$, $\varrho = (\varrho_1, \varrho_2, ..., \varrho_n)$, $\mathbf{1}$ is an all-1 vector, and $< \cdot, \cdot >$ denotes the Frobenius product of two matrices.

Furthermore, Eq. (9) can be rewritten as the smallest Kullback-Leibler (KL) divergence [19], [37];

$$W_p^p(\mathbf{D}, \mathbf{\Pi}) = \epsilon \min_{\mathbf{\Pi}} \mathrm{KL}(\mathbf{\Pi}|\xi), \tag{10}$$

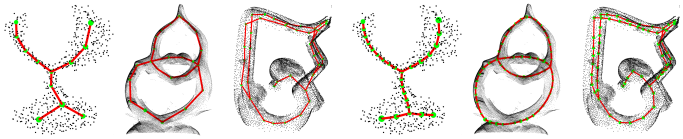
from $\mathbf{\Pi}$ to the distance-based kernel $\xi = e^{-\frac{\mathbf{D}}{\epsilon}}$. The exponential is computed component-wise. Eq. (10) provides a new interpretation for the regularized transportation problem: the optimal plan π is the projection of the distance-based kernel ξ. The KL divergence is then defined as:

$$\mathrm{KL}(\mathbf{\Pi}|\xi) = \sum_i \sum_j \Pi_{ij} \big(log(\frac{\Pi_{ij}}{\xi_{ij}}) - 1 \big).$$

This minimization is convex. The optimal transport plan $\mathbf{\Pi}$ is the projection of the distance-based kernel ξ onto $\mathbf{\Pi}$, and can be obtained by iterative Bregman projection [37], [39].

$$\mathbf{\Pi} = diag(\mathbf{u}) \xi diag(\mathbf{v}), \tag{11}$$

where $\mathbf{u} \in R_+^n$, and $\mathbf{v} \in R_+^m$. Further details can be found in [36], [37] and [39].



(a) Rough MdCS (b) Smooth MdCS

Fig. 4: Extracted MdCSs on different models.

Suppose $\mathbf{\Pi}$ is fixed, we use an efficient quasi-Newton solver to compute X. The gradient of $W_p^p (p = 1)$ is expressed as:

$$\begin{aligned} \frac{\partial W_p^p}{\partial x_i} = & \sum_{j=1}^{n} \frac{x_i - y_j}{\|x_i - y_j\|} \Pi_{ij} + 2\lambda \left(x_i - \frac{1}{v(x_i)} \sum_{x_k \in N(x_i)} x_k \right) \\ & - 2\lambda \sum_{x_i \in N(x_l), v(x_l) > 1} \frac{1}{v(x_l)} \left(x_l - \frac{1}{v(x_l)} \sum_{x_k \in v x_l} x_k \right). \end{aligned} \tag{12}$$

where $v(x_i)$ is the number of one-ring neighbors of x_i. Let $\frac{\partial W_p^p}{\partial x_i} = 0$, system of equations with x_i can be written as:

$$Ax_i = -B, \tag{13}$$

where

$$A = \begin{cases} C + 2\lambda D + 2\lambda & v(x_i) > 1, \\ C + 2\lambda D & others, \end{cases}$$

and

$$B = \begin{cases} F - 2\lambda G - \frac{2\lambda}{v(x_i)} \sum\limits_{x_k \in N(x_i)} x_k & v(x_i) > 1, \\ F - 2\lambda G & others. \end{cases}$$

Here, C, D, F and G are represented as $C = \sum\limits_{j=1}^{n} \frac{\Pi_{ij}}{\|x_i - y_j\|}$, $D = \sum\limits_{x_i \in N(x_l), v(x_l) > 1} \frac{1}{v^2(x_l)}$, $F = \sum\limits_{j=1}^{n} \frac{y_j}{\|x_i - y_j\|} \Pi_{ij}$, and

$$G = \sum_{x_i \in x_l, v(x_l) > 1} \frac{1}{v(x_l)} \left(x_l - \frac{1}{v(x_l)} \sum_{x_k \in q_l, x_k \neq x_i} x_k \right).$$

By applying a fixed point iteration, x_i is updated as:

$$x_i = (1 - \theta) x_i - \theta B / A, \quad 0 < \theta < 1. \tag{14}$$

V. Result and Analysis

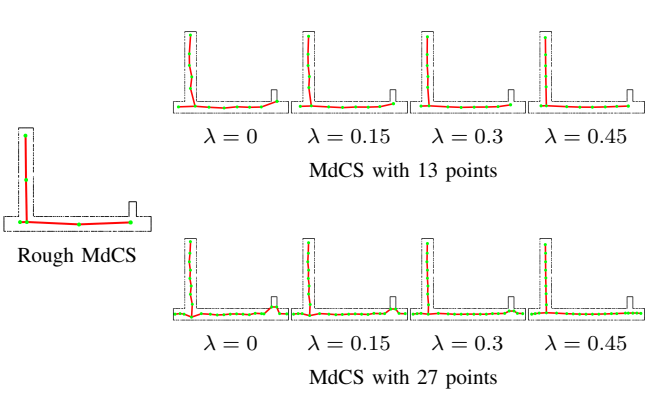
We demonstrate the effectiveness of MdCSs extraction on various point clouds, such as complete and clean point clouds, point clouds with large gap holes, and raw scan point clouds (Fig. 4). We also compare our results with state-of-the-art methods. In our experiments, ϱ_j is initiated in terms of sampling density. We initially compute the minimum radius r_j of k-nearest neighbors for every point y_j and use r_j^2 as an original ϱ_j. Then, the final ϱ_j is obtained via normalization. In all the figures, we use the size of points to represent the mass of skeleton points. All the models used in this work are listed in Table IV.

TABLE II: Models. n is the number of original sampling points of the model. We use ✓ to indicate that the model has holes.

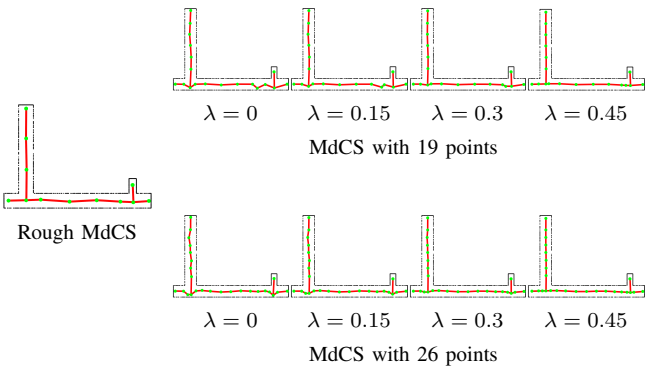
Fig.	1	4			10		11		12	14	15			16			
Model	Deer	Leaf	Couple	IndiaLady	Wolf	Man-3	Y	Man-1	Man-2	Dumbbell	Gecko	Dancers	RecSkirt	Scan-1	Scan-2	Scan-3	Scan-4
n	22568	579	43545	8690	4344	3252	7029	3252	2231	736	14182	15585	21358	18831	17518	18086	18668
Holes	✓		✓	✓							✓	✓	✓				

TABLE IV: Number of topology errors with respect to different point cloud artifacts.

Method	Model																													
	Man-1										Man-3										Wolf									
		Noise(σ)			Missing(r)			Missing(%)				Noise(σ)			Missing(r)			Missing(%)				Noise(σ)			Missing(r)			Missing(%)		
		1.5	2.5	3.5	1.5	1.8	2.0	20	40	50		1.5	2.5	3.5	1.5	1.8	2.0	20	40	50		1.5	2.5	3.5	1.5	1.8	2.0	20	40	50
ℓ_1	0	1	2	2	1	1	2	1	1	1	0	0	0	0	0	0	2	0	0	0	2	2	1	2	1	1	2	1	2	1
LBC	0	0	1	2	1	1	1	1	1	1	0	0	0	2	0	0	1	0	0	0	0	0	2	3	0	0	1	0	0	0
$ROSA$	1	1	1	2	0	0	2	1	0	1	0	0	1	0	1	1	1	0	0	0	0	1	1	4	1	1	2	0	0	1
$MdCS$	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	2	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**

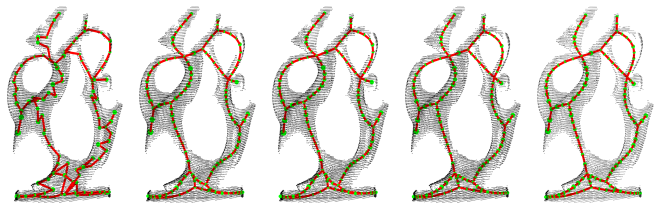


(a) Using 10 initiated points to extract an MdCS



(b) Using 20 initiated points to extract an MdCS

Fig. 5: Comparison of the MdCS generated using different numbers of initiated points, different numbers of final skeleton points, and different values of parameter λ.



(a) $\lambda = 0$ (b) $\lambda = 0.15$ (c) $\lambda = 0.3$ (d) $\lambda = 0.6$ (e) $\lambda = 0.9$

Fig. 6: Comparison of the different values of the parameter λ.

TABLE III: Default settings used for user controllable parameters.

Parameter	m/n	ϵ	$\Delta\epsilon$	λ	θ	τ
Default value	0.1	0.001	0.0002	0.3	0.1	0.15

A. *Impact of parameters.*

Several parameters are adopted in our algorithm. The parameter ϵ plays the role of controling the intersection set of $Y_i \cap Y_k$. The larger ϵ is, the larger numbers of points in the intersection sets $Y_i \cap Y_k$ are. Furthermore, ϵ represents the scale of the curve skeleton, and is updated step by step till there is no any non-skeleton point. The parameter τ is a threshold, which is used to test the symmetry of the transport plan π_i. The smaller τ, more possible the π is tested as non-symmetry. The parameter θ is used to compute the position of points x_i iteratively. The larger θ is, the faster the convergence rate is. Nonetheless, an excessively θ may lead to divergence. The parameters τ and θ are fixed in our algorithm. The parameters m and λ are free parameters, and play important roles. The

(a) Using a few initiated curve skeleton points

(b) Using numerous initiated curve skeleton points

Fig. 7: Impact of the different numbers of initiated skeleton points.

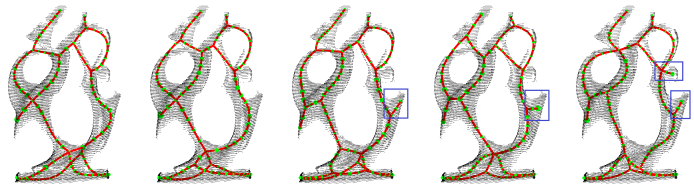
default parameter sets listed in Table II are applied throughout all the presented experiments, except for the example demonstrated in Figs. 5, 6, and 7, in which we demonstrate the influence of the change of parameters m and λ.

Figs. 5 and 6 demonstrate the influence of the parameter λ. A high value of λ tends to obtain a line branch of a curve skeleton. Unwanted oscillations occur avoidable when too more points are inserted into topology graphs. With the increase in λ, the MdCS tends to converge to line segments.

In Fig. 5, we demonstrate the role of the parameters m. The initiated m plays a role to control the smallest scale of an MdCS. In Fig. 5[b], a high number of initiated curve skeleton points indicates that the extracted curve skeleton can capture further geometry details of point clouds. In general, a satisfactory MdCS can be obtained when $m = \lfloor 0.1n \rfloor$. For the RecSkirt displayed in Fig 15, the initiated m is set to be $\lfloor 0.2n \rfloor$ because the RecSkirt includes small scale structure. In Fig. 7, we apply a different parameter m to extract an MdCS from point clouds with large gap holes. To highlight the difference in MdCSs on different values of m, only locally enlarged MdCSs are presented in Fig. 7. Whole MdCSs with larger value of m are shown in Fig. 15. The parameter m is less in Fig. 7[a] than in Fig. 7[b] for every model. We can find that certein thin structures in the MdCS are missed in Fig. 7[a] in contrast to the MdCS demonstrated in Fig. 7[b]. Furthermore, we evaluate the stability of our algorithm over complex data with large gaping holes exhibited in Fig. 8 by setting different points with the same number as the initiation of skeleton points. In these experiments, $m = \lfloor 0.1n \rfloor$. While the main topology structures are preserved, various details of the curve skeletons may be different under various initiations of skeleton points. For example, the labeled small branch displayed in Figs. 8[c-e] are lost in Figs. 8[a-b]. To capture further details of point clouds, m must be initiated as a high value, rather than a low value. However, additional computational cost will be taken if additional initiated curve skeleton points are selected.

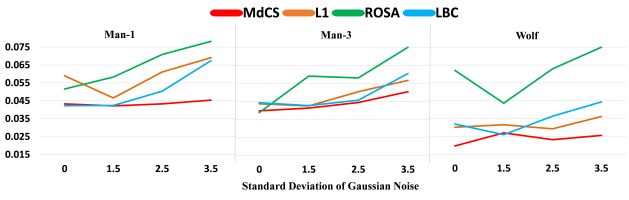
B. *Quantification analysis.*

In Figs. 9 and 10 and Table III, we make a quantitative analysis of the extracted curve skeleton from point clouds with d-



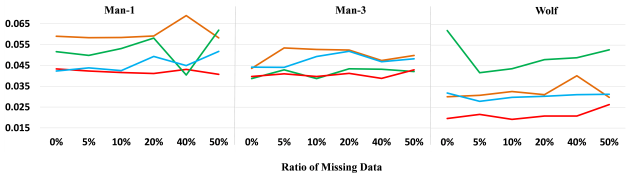
(a) (b) (c) (d) (e)

Fig. 8: Comparison of curve skeletons on different initiation points with the same number of points.



(a) Noise(σ)

(b) Missing(r)



(c) Missing(%)

Fig. 9: Comparison of the Haudorff distance between a ground truth curve skeleton and the extracted curve skeleton with respect to different point clouds artifacts and various methods.

ifferent artifacts. Mean curvature skeleton (MCS) method [40] is used to extract a curve skeleton from the mesh model, which is reconstructed from clean point clouds. The extracted curve skeleton from the mesh model considered as the ground truth of the curve skeleton of the corresponding point clouds. In our experiments, clean point clouds is perturbed via random Gaussian noise or two kinds of missing data. In Gaussian noise perturbations, the standard deviation of a Gaussian function is σd_{min}, and d_{min} is the minimum distance between points in point clouds. Gaussian noise perturbations are represented as Noise(σ). Missing data are generated by two methods. In the first one, the points within a ball are deleted from the original point clouds. The center of the ball is a random point in the original point clouds, and the radius of the ball is the r times of the minimum distance from the random point to the curve skeleton extracted from the mesh model. The first missing data are presented as Missing(r). In the second one, several points are randomly deleted from the original point clouds, and the second missing data are presented as Missing(%). In

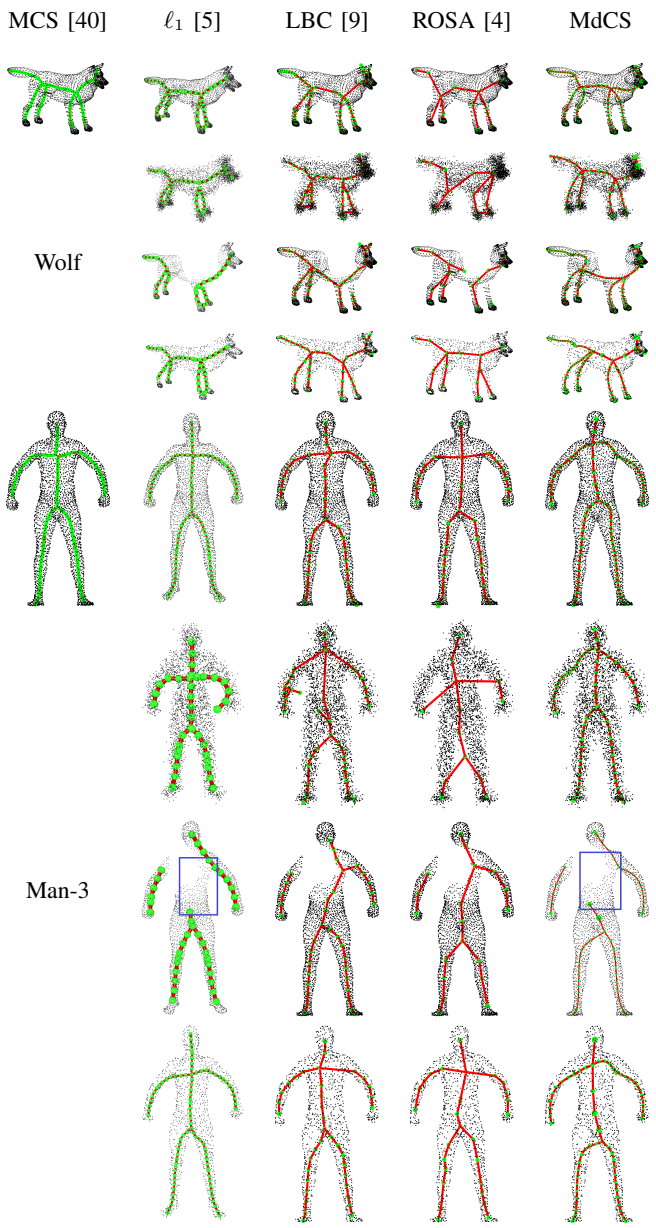


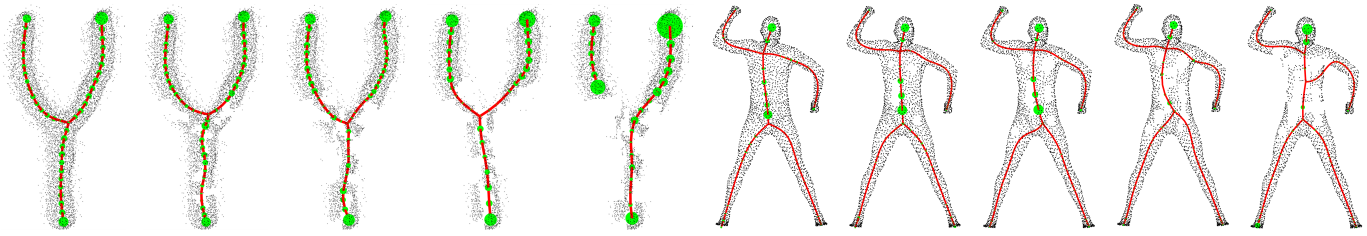
Fig. 10: Skeleton extraction comparisons of point clouds with different artifacts. The first and fifth rows show the extracted curve skeleton from the point clouds without artifacts. The extracted curve skeletons from the point clouds with Noise ($\sigma = 3.5$), Missing ($r = 2$) and Missing (50%) levels are shown in the second and sixth, third and seventh, fourth and eighth rows, correspondingly. The curve skeleton shown on the left is the ground truth.

Fig. 9, we compute the Hausdorff distance [41] between the extracted curve skeleton from the point clouds and the ground truth of the curve skeleton of the corresponding model. Under the cases of the three kind of point cloud artifacts, we can find that the curve skeletons extracted in our algorithm are the closer to the ground truth than those extracted from other methods, namely, LBC [9], ROSA [4] and ℓ_1-medial [5], on the Hausdorff distances. In Table III, we count the number of wrong topology connections of the extracted curve skeleton. In 27 curve skeletons extracted from 27 point cloud data, there are $13, 17, 19$ and 1 curve skeletons, in which the incorrect topology connection exists, and separately extracted from LBC [9], ROSA [4], ℓ_1-medial [5], and MdCS. Furthermore, we analyze the robustness of our algorithm under different point cloud artifacts. In the Fig. 9, the Hausdorff distance changes slightly with the increase of the standard deviation of Gaussion noise or the ratio of random missing data, and the Hasudorff distance changes abruptly when the radius of the missing ball changes from 1.8 to 2. This result shows that our algorithm performs more robustly for the case of Gaussian noise and sparse sampling than for the case of Missing(r). The skeleton branches may be far from the ground truth of the curve skeleton of the model for the case of Missing(r). The main reason is that the skeleton points is defined as the center of the local power cell, which depends on the local point distribution, other than the geometry center of the local shape. When the radius of the missing ball is equal to 2, there are two errors of the topology structure of the model Man-3 in the curve skeletons extracted in the ℓ_1-medial and MdCS methods, and only one error in the LBC and ROSA methods (Fig. 9). The reason is that the ℓ_1-medial and MdCS methods depend on the initiation of skeleton points. Only when sufficient points are sampled as the initiation in the blue box in the ℓ_1-medial and MdCS methods, the topology connection at the blue box can be reconstructed as the curve skeleton extracted in the LBC and ROSA methods.

In Fig. 11, we compare the results of the incomplete models generated via interactive manipulations step by step. When small areas of missing data exist, the correct MdCS can be extracted. If a large area of missing data exists, the topology connection can be changed, as illustrated in Figs. 11[e] and 11[j]. In Fig. 12, the incorrect MdCS may be generated with the increase of σ of Gaussian noise.

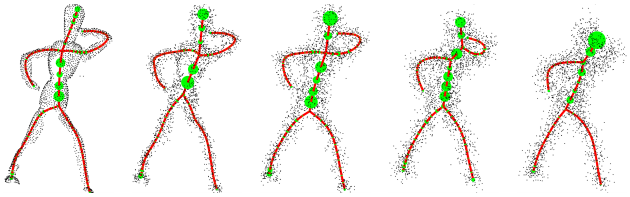
C. Non-uniform sampling.

The robustness of MdCS extraction from non-uniform point clouds is demonstrated in Fig. 13. The original point cloud Y is represented as a non-uniform probability distribution in according with the local sampling density, and an initial skeleton point set X is still represented as a uniform probability distribution. From the experiments, the extracted MdCSs from non-uniform point clouds can accurately match with the curve skeletons from the corresponding uniform point clouds, even when the ratio of the local point density reaches 1:7. The mass property of the initiated curve skeleton points plays an important role. Although random sampling is applied to initiate curve skeleton points, a uniform distribution of these



(a) $A = 0$ (b) $A = 10\%$ (c) $A = 20\%$ (d) $A = 31\%$ (e) $A = 36\%$ (f) $A = 0$ (g) $A = 5\%$ (h) $A = 10\%$ (i) $A = 16\%$ (j) $A = 25\%$

Fig. 11: Impact of the different levels of missing data on Models Y and Man-1. Rough and smooth MdCSs are presented in the first and second rows, respectively. (a) and (f) are the original point clouds without missing data. (b-e) and (g-k) are point clouds with missing data. The value of A denotes the percentage of missing data.



(a) (b) $\sigma = 1.0$ (c) $\sigma = 1.8$ (d) $\sigma = 2.8$ (e) $\sigma = 3.0$

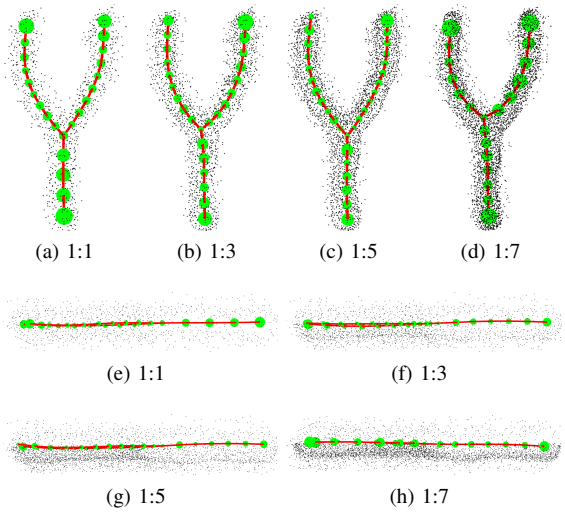
Fig. 12: Impact of the different levels of noise on Model Man-2. Rough and smooth MdCSs are exhibited in the first and second rows, respectively. (a) is the original point clouds. (b-e) are the point clouds corrupted by Gaussian noise with zero-mean and standard deviation σ.

points is initially obtained in Line 9 of Algorithm 1 before these points are merged because the probability density is the same for every initiated curve skeleton point. Moreover, we consider the non-uniform probability distribution of $\mathbf{Y}$. Thus, our algorithm is feasible for non-uniform sampling points $\mathbf{Y}$.

D. Comparison.

In Fig. 14, we use the codes provided by the authors in [5] to extract ℓ_1-medial skeleton from a *Dumbbell* model using different parameters. The MdCS accurately captures the structure of the *Dumbbell*, better than the generated ℓ_1-medial skeleton. The possible reason is that the supporting neighborhood radius, used in the definition of ℓ_1-medial skeleton, must be gradually increased to contract non-branch points to produce skeleton branches. Nonetheless, this key radius may require a sudden change from the rod to the bell for the *Dumbbell*-shape model. The ℓ_1-medial skeleton cannot catch the local centers of the road and two bells well. In our algorithm, topology checking provides a strategy to find the different scale structures and the structure with holes.

In Figs. 15 and 16, we separately make comparisons between our algorithm and LBC [9], ROSA [4], and ℓ_1-medial [5] on point clouds with large gap holes and raw scan point clouds. For point clouds with large gap holes depicted in Fig. 15, the curve skeletons from our algorithm contain fewer geometrical and topological errors than the curve skeletons from LBC [9], ROSA [4], and ℓ_1-medial [5]. For raw scan point clouds from a Kinect displayed in Fig. 16, the curve



(a) 1:1 (b) 1:3 (c) 1:5 (d) 1:7

(e) 1:1 (f) 1:3

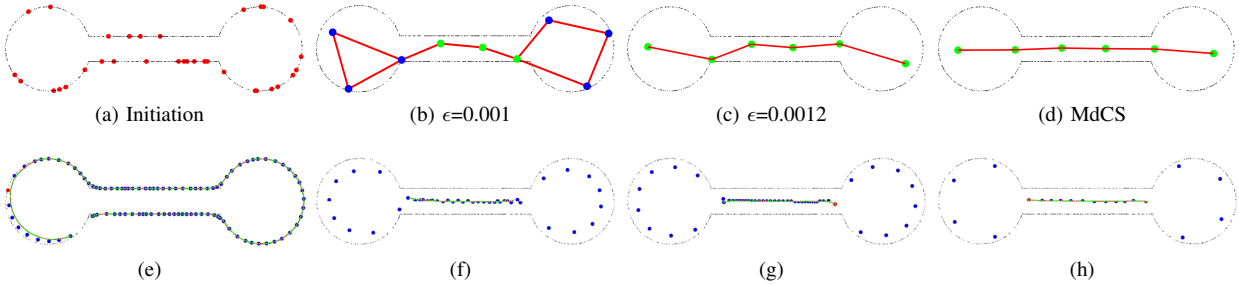
(g) 1:5 (h) 1:7

Fig. 13: Non-uniform sampling. We evaluate the efficiency of an MdCS on non-uniform sampling by comparing with that of on uniform sampling (a). Ratio of non-uniform sampling is 1:3 (b), 1:5 (c), and 1:7 (d), correspondingly. We show the side views of (a-d) in (f-j) on the right.

skeletons from our algorithm can accurately capture geometry and topology properties of the point clouds. The curve skeletons from LBC [9], ROSA [4], and ℓ_1-medial [5] may contain several incorrect branches or lost certain branches.

E. Complexity and performance.

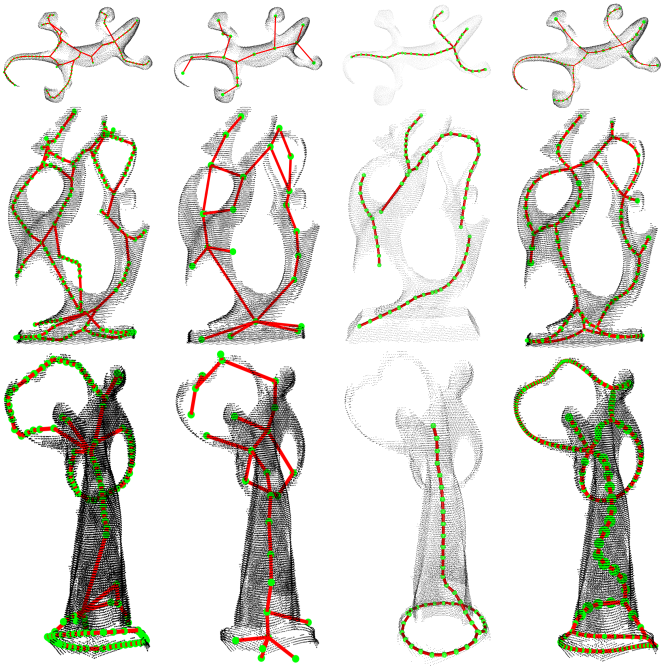
Our algorithm has been developed within the MATLAB environment. Thus the speed of execution is not optimized. In Table V, we analyze the computational complexity of the single steps of the algorithm. The overall computational cost is $O(n^2 log(n))$. The most time-consuming step is the computation of $\mathbf{\Pi}$ via iterative Bregman projection, which is in the step of topology extraction, and the computational complexity of the computation of $\mathbf{\Pi}$ is $O(n^2)$. We also compare our algorithm with other methods in terms of running time listed in Table VI. The performance is measured on an Intel Core i5-4590 @ 3.30 GHz with 8.00 GB RAM. In the future, we may develop or use another fast method for



(a) Initiation (b) ϵ=0.001 (c) ϵ=0.0012 (d) MdCS

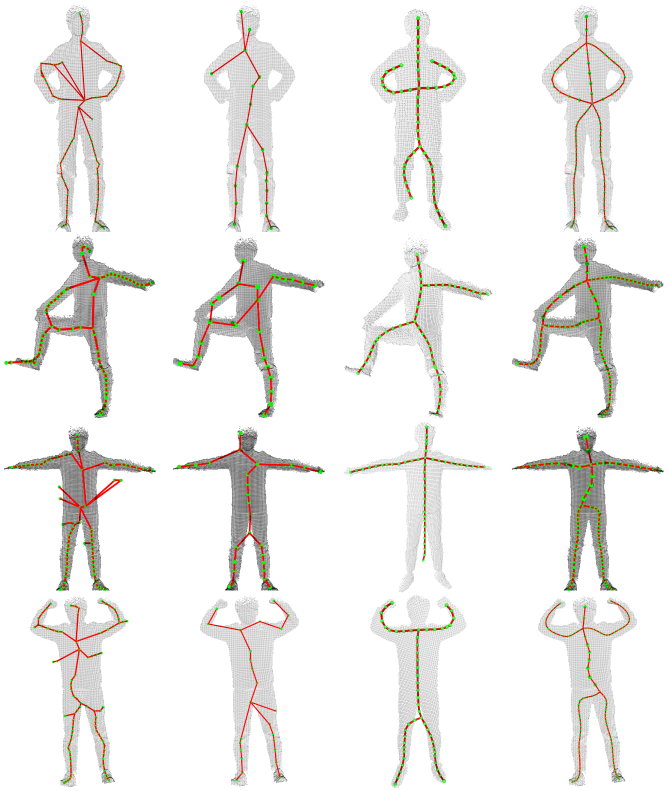
(e) (f) (g) (h)

Fig. 14: Comparison of MdCS and ℓ_1-medial skeleton [5] on a *Dumbbell* model. (a-d) are the results from our algorithm. (e-h) are the ℓ_1-medial skeletons [5] with different parameters. In (e-h), the blue points are possible skeleton points, and the green points are final skeleton points.



(a) LBC [9] (b) ROSA [4] (c) ℓ_1-medial [5] (d) Ours

Fig. 15: Skeleton extraction comparisons of point clouds that are highly incomplete, thus containing large gaping holes.



(a) LBC [9] (b) ROSA [4] (c) ℓ_1-medial [5] (d) Ours

Fig. 16: Skeleton extraction comparisons of various raw scan point clouds of human.

computing $\mathbf{\Pi}$, such as fast proximal point method [42], to relax the computational complexity of our algorithm further.

F. Limitations.

Skeletonization from point clouds is generally an ill-posed problem, especially with the existence of missing data. If the amount of noise or missing data are too large, our algorithm may produce erroneous outputs (e.g., Fig. 11[j] and Fig. 12[j]). We also notice that certain fine-scale structures may be missed (Fig. 7[a]) when the amount of missing data is too large, and the number of initiated curve skeleton points is too small. To capture a complete curve skeleton from point clouds with large missing data, numerous initiated curve skeleton points must be used, as illustrated in Fig. 7[b], which will increase the computational cost of our algorithm. Although our algorithm

TABLE V: Computational cost of the main steps of the proposed approach, where n is the number of original point clouds, m, s, and t denote the number of original, rough, and final curve skeleton points, respectively.

Task	Computational cost
Load	$O(n)$
Topology Extraction	$O(nmlog(m-s))$
Topology Checking	$O(n+s)$
Geometry Optimization	$O(nt)$

TABLE VI: Comparison of running times.

Model	n	m	Times(s)			
			LBC	ℓ_1	ROSA	MdCS
Man-1	3252	300	18	15	61	50
Man-3	3252	300	18	18	61	48
Gecko	14182	500	103	34	21654	661
Dancers	15585	1500	189	36	14708	676
RecSkirt	21358	4200	234	53	29275	1360

performs better than ROSA [9] in terms of computational cost, our algorithm performs worse than ℓ_1-medial skeleton [5]. The performance of our algorithm must be improved in the future.

VI. Conclusion

With the Wasserstein distance, we introduce an MdCS as a novel abstraction representation for point clouds. By using an entropic regularization term, topology extraction and geometric optimization of a curve skeleton from point clouds are modeled as a unified framework, thereby minimizing the Wasserstein distance between point clouds and its curve skeleton. Experiments clearly demonstrate that our proposed algorithm performs robustly on unorganized, unoriented, and highly incomplete point cloud data.

In the future, we aim to extend an MdCS to represent the principal curves of high-dimensional data [43]. It is also quite interesting to explore the ability of the optimal mass transport to capture the relationship between surfaces and their MdCSs, similar to that [44]. Furthermore, our goal is to improve the efficiency of the current framework for large-scale point clouds and explore the possibility of using an MdCS for the robust surface reconstruction [45] from low-quality point clouds.

Acknowledgment

We thank anonymous reviewers and the editor for their valuable and constructive suggestions. This work was supported in parts by NSFC (61772097, 61761146002, 61861130365), 973 Program (2015CB352501), Chongqing Research Program of Application Foundation and Advanced Technology (cstc2018jcyjA0604), Guangdong Science and Technology Program (2015A030312015), and Shenzhen Innovation Program (KQJSCX20170727101233642).

References

[1] H. Blum, "A Transformation for Extracting New Descriptors of Shape," *Models for the Perception of Speech and Visual Form*, pp. 362–380, 1967.

[2] K. Siddiqi and S. Pizer, *Medial representations: mathematics, algorithms and applications*. Springer Science & Business Media, 2008.

[3] N. D. Cornea, D. Silver, and P. Min, "Curve-skeleton properties, applications, and algorithms," *IEEE Transactions on Visualization and Computer Graphics*, vol. 13, no. 3, pp. 0530–548, 2007.

[4] A. Tagliasacchi, H. Zhang, and D. Cohen-Or, "Curve skeleton extraction from incomplete point cloud," *ACM Transactions on Graphics (TOG)*, vol. 28, no. 3, pp. 71:1–71:10, 2009.

[5] H. Huang, S. Wu, D. Cohen-Or, M. Gong, H. Zhang, G. Li, and B. Chen, "L1-medial skeleton of point cloud." *ACM Transactions on Graphics (TOG)*, vol. 32, no. 4, pp. 65:1–65:8, 2013.

[6] A. Tagliasacchi, T. Delame, M. Spagnuolo, N. Amenta, and A. Telea, "3D skeletons: A state-of-the-art report," *Computer Graphics Forum*, vol. 35, no. 2, pp. 573–597, 2016.

[7] R. Ogniewicz and M. Ilg, "Voronoi skeletons: Theory and applications," in *IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 1992.* IEEE, 1992, pp. 63–69.

[8] A. Sharf, T. Lewiner, A. Shamir, and L. Kobbelt, "On-the-fly curve-skeleton computation for 3d shapes," *Computer Graphics Forum*, vol. 26, no. 3, pp. 323–328, 2007.

[9] J. Cao, A. Tagliasacchi, M. Olson, H. Zhang, and Z. Su, "Point cloud skeletons via laplacian based contraction," in *Shape Modeling International Conference (SMI), 2010*. IEEE, 2010, pp. 187–197.

[10] O. K.-C. Au, C.-L. Tai, H.-K. Chu, D. Cohen-Or, and T.-Y. Lee, "Skeleton extraction by mesh contraction," *ACM Transactions on Graphics (TOG)*, vol. 27, no. 3, pp. 44:1–44:10, 2008.

[11] Y. Livny, F. Yan, M. Olson, B. Chen, H. Zhang, and J. El-Sana, "Automatic reconstruction of tree skeletal structures from point clouds," *ACM Transactions on Graphics (TOG)*, vol. 29, no. 6, pp. 151:1–151:8, 2010.

[12] G. Li, L. Liu, H. Zheng, and N. J. Mitra, "Analysis, reconstruction and manipulation using arterial snakes," *ACM Transactions on Graphics(TOG)*, vol. 29, no. 6, pp. 152:1–152:10, 2010.

[13] A. Bucksch, R. Lindenbergh, and M. Menenti, "Skeltre:robust skeleton extraction from imperfect point clouds," *The Visual Computer*, vol. 26, no. 10, pp. 1283–1300, 2010.

[14] M. Natali, S. Biasotti, G. Patanè, and B. Falcidieno, "Graph-based representations of point clouds," *Graphical Models*, vol. 73, no. 5, pp. 151–164, 2011.

[15] W. Ju, D. Xiang, B. Zhang, L. Wang, I. Kopriva, and X. Chen, "Random walk and graph cut for co-segmentation of lung tumor on pet-ct images," *IEEE Transactions on Image Processing*, vol. 24, no. 12, pp. 5854–5867, 2015.

[16] C. Song, Z. Pang, X. Jing, and C. Xiao, "Distance field guided l1-median skeleton extraction," *The Visual Computer*, 2016.

[17] X. Zhang, G. Bao, W. Meng, M. Jaeger, H. Li, O. Deussen, and B. Chen, "Tree branch level of detail models for forest navigation," in *Computer Graphics Forum*, vol. 36, no. 8. Wiley Online Library, 2017, pp. 402–417.

[18] L. Lu, B. Lévy, and W. Wang, "Centroidal voronoi tessellation of line segments and graphs," *Computer Graphics Forum*, vol. 31, no. 2, pp. 775–784, 2012.

[19] J. Solomon, F. De Goes, G. Peyré, M. Cuturi, A. Butscher, A. Nguyen, T. Du, and L. Guibas, "Convolutional wasserstein distances: Efficient optimal transportation on geometric domains," *ACM Transactions on Graphics (TOG)*, vol. 34, no. 4, pp. 66:1–66:11, 2015.

[20] Q. Mérigot, J. Meyron, and B. Thibert, "An algorithm for optimal transport between a simplex soup and a point cloud," *SIAM Journal on Imaging Sciences*, vol. 11, no. 2, pp. 1363–1389, 2018.

[21] T. Ju, S. Schaefer, and J. Warren, "Mean value coordinates for closed triangular meshes," *ACM Transactions on Graphics (TOG)*, vol. 24, no. 3, pp. 561–566, 2005.

[22] F. d. Goes, D. Cohen-Steiner, P. Alliez, and M. Desbrun, "An Optimal Transport Approach to Robust Reconstruction and Simplification of 2D Shapes," *Computer Graphics Forum*, vol. 30, pp. 1539–1602, 2011.

[23] J. Digne, D. Cohen-Steiner, P. Alliez, F. Goes, and M. Desbrun, "Feature-preserving surface reconstruction and simplification from defect-laden point sets," *J. Math. Imaging Vis.*, vol. 48, no. 2, pp. 369–382, 2014.

[24] F. de Goes, K. Breeden, V. Ostromoukhov, and M. Desbrun, "Blue noise through optimal transport," *ACM Transactions on Graphics (TOG)*, vol. 31, no. 6, p. 171, 2012.

[25] H. Qin, Y. Chen, J. He, and B. Chen, "Wasserstein blue noise sampling," *ACM Transactions on Graphics (TOG)*, vol. 36, no. 5, pp. 168:1–168:13, 2017.

[26] C. Villani, *Optimal transport: old and new*. Springer Science & Business Media, 2008.

[27] F. Aurenhammer, F. Hoffmann, and B. Aronov, "Minkowski-type theorems and least-squares clustering," *Algorithmica*, vol. 20, no. 1, pp. 61–76, 1998.

[28] N. Amenta, S. Choi, and R. K. Kolluri, "The power crust, unions of balls, and the medial axis transform," *Computational Geometry*, vol. 19, no. 2-3, pp. 127–153, 2001.

[29] N. Amenta, S. Choi, and R. K. Kolluri, "The power crust," in *Proceedings of the sixth ACM symposium on Solid modeling and applications*, 2001, pp. 249–266.
[30] A. Telea and A. Jalba, "Computing curve skeletons from medial surfaces of 3d shapes." in *Proc. Theory and Practice of Computer Graphics (TPCG)*, 2012, pp. 99–106.
[31] A. C. Jalba, A. Sobiecki, and A. C. Telea, "An unified multiscale framework for planar, surface, and curve skeletonization," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 38, no. 1, pp. 30–45, 2016.
[32] A. C. Jalba, J. Kustra, and A. C. Telea, "Surface and curve skeletonization of large 3d models on the GPU," *IEEE Transactions on Pattern Analysis & Machine Intelligence*, no. 6, pp. 1495–1508, 2013.
[33] D. Reniers, J. Van Wijk, and A. Telea, "Computing multiscale curve and surface skeletons of genus 0 shapes using a global importance measure," *IEEE Transactions on Visualization and Computer Graphics*, vol. 14, no. 2, pp. 355–368, 2008.
[34] U. Ozertem and D. Erdogmus, "Locally defined principal curves and surfaces," *Journal of Machine Learning Research*, vol. 12, no. 4, pp. 1249–1286, 2011.
[35] F. Aurenhammer, "Power diagrams: Properties, algorithms and applications," *SIAM Journal on Computing*, vol. 16, no. 1, pp. 78–96, 1987.
[36] M. Cuturi and A. Doucet, "Fast computation of Wasserstein barycenters," in *Proceedings of The 31st International Conference on Machine Learning*, 2014, pp. 685–693.
[37] M. Cuturi, "Sinkhorn distances: Lightspeed computation of optimal transport," in *Advances in Neural Information Processing Systems*, 2013, pp. 2292–2300.
[38] T. Hastie and W. Stuetzle, "Principal curves," *Journal of the American Statistical Association*, vol. 84, no. 406, pp. 502–516, 1989.
[39] J.-D. Benamou, G. Carlier, M. Cuturi, L. Nenna, and G. Peyré, "Iterative Bregman projections for regularized transportation problems," *SIAM Journal on Scientific Computing*, vol. 37, no. 2, pp. 1111–1138, 2015.
[40] A. Tagliasacchi, I. Alhashim, M. Olson, and H. Zhang, "Mean curvature skeletons," in *Computer Graphics Forum*, vol. 31, no. 5. Wiley Online Library, 2012, pp. 1735–1744.
[41] F. Mémoli and G. Sapiro, "A theoretical and computational framework for isometry invariant recognition of point cloud data," *Foundations of Computational Mathematics*, vol. 5, no. 3, pp. 313–347, 2005.
[42] Y. Xie, X. Wang, R. Wang, and H. Zha, "A fast proximal point method for computing wasserstein distance," *arXiv:1802.04307*, 2018.
[43] S. Wu, P. Bertholet, H. Huang, D. Cohen-Or, M. Gong, and M. Zwicker, "Structure-aware data consolidation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2018.
[44] S. Wu, , H. Huang, M. Gong, M. Zwicker, and D. Cohen-Or, "Deep points consolidation," *ACM Transactions on Graphics (TOG)*, vol. 34, no. 6, pp. 176:1–176:13, 2015.
[45] M. Berger, A. Tagliasacchi, L. M. Seversky, P. Alliez, G. Guennebaud, J. A. Levine, A. Sharf, and C. T. Silva, "A survey of surface reconstruction from point clouds," *Computer Graphics Forum*, vol. 36, no. 1, pp. 301–329, 2017.

Jia Han is a Master's student in the College of Computer Science and Technology at Chongqing University of Posts and Telecommunications. His research interests include point clouds and image processing.

Ning Li is a Master's student in the College of Computer Science and Technology at Chongqing University of Posts and Telecommunications. His research interests include point clouds and image processing.

Hui Huang is now a Distinguished Professor of Shenzhen University, where she directs the Visual Computing Research Center at College of Computer Science and Software Engineering. She received her PhD in Applied Math from The University of British Columbia in 2008 and another PhD in Computational Math from Wuhan University in 2006. Her research interests mainly lie on Computer Graphics and Computer Vision. She is a Senior Member of IEEE and ACM, a Distinguished Member of CCF, an Associate Editor-in-Chief of The Visual Computer and is on the editorial board of Computers & Graphics.

Hongxing Qin is now a Professor at Chongqing University of Posts and Telecommunications, Chongqing. He received his PhD degree in pattern recognition from Shanghai Jiaotong University, in 2008. He worked as a postdoctoral researcher at Rutgers, the State University of New Jersey, from 2008 to 2009. His research interests include computer graphics, digital geometry processing, medical image processing, and visualization.

Baoquan Chen is now a Professor and Executive Director of Center for Frontier of Computing Study at Peking University. He received his MS degree from Tsinghua University, Beijing, and PhD degree from the State University of New York at Stony Brook. He is the recipient of the NSF CAREER award 2003, IEEE Visualization Best Paper Award 2005, and NSFC Outstanding Young Researcher program in 2010. His research interests generally lie in computer graphics, visualization, and human-computer interaction.