L_1-Medial Skeleton of Point Cloud

Hui Huang[⋆∗] Shihao Wu[⋆†] Daniel Cohen-Or[‡] Minglun Gong[§] Hao Zhang[✻] Guiqing Li[†] Baoquan Chen[⋆∗]
[⋆]Shenzhen VisuCA Key Lab / SIAT [†]South China University of Technology
[‡]Tel-Aviv University [§]Memorial University of Newfoundland [✻]Simon Fraser University

Abstract

We introduce L_1-*medial skeleton* as a curve skeleton representation for 3D point cloud data. The L_1-median is well-known as a robust global center of an arbitrary set of points. We make the key observation that adapting L_1-medians *locally* to a point set representing a 3D shape gives rise to a one-dimensional structure, which can be seen as a localized center of the shape. The primary advantage of our approach is that it does not place strong requirements on the quality of the input point cloud nor on the geometry or topology of the captured shape. We develop a L_1-medial skeleton construction algorithm, which can be directly applied to an unoriented raw point scan with significant noise, outliers, and large areas of missing data. We demonstrate L_1-medial skeletons extracted from raw scans of a variety of shapes, including those modeling high-genus 3D objects, plant-like structures, and curve networks.

CR Categories: I.3.5 [Computer Graphics]: Computational Geometry and Object Modeling—[Curve, surface, solid, and object representations]

Keywords: curve skeleton, point cloud, L_1-median, regularization

Links: DL PDF WEB VIDEO DATA CODE

1 Introduction

L_1-median is a simple and powerful statistical tool that extends the univariate median to the multivariate setting [Weber 1909]. It represents a unique *global center* of a given set of points [Daszykowski et al. 2007], with the prominent property that it is robust to outliers and noise. In this paper, we make the key observation that adapting L_1-medians *locally*, instead of globally, to a set of points representing a geometric shape, gives rise to a *one-dimensional* structure. The structure can be seen as a localized center of the shape, i.e., a medial curve skeleton. We introduce such a structure which we call the L_1-*medial skeleton*; it amounts to a spatially localized version of the L_1-median with conditional regularization.

Given an unorganized and unoriented set of points $Q = \{q_j\}_{j\in J} \subset \mathbb{R}^3$, we investigate the following definition for L_1-medial skeletons that leads to an optimal set of projected points $X = \{x_i\}_{i\in I}$:

$$\underset{X}{\operatorname{argmin}} \sum_{i\in I}\sum_{j\in J} \|x_i - q_j\|\theta(\|x_i - q_j\|) + R(X), \qquad (1)$$

∗Corresponding authors: Hui Huang (hhzhiyan@gmail.com), Baoquan Chen (baoquan.chen@gmail.com)

ACM Reference Format
Huang, H., Wu, S., Cohen-Or, D., Gong, M., Zhang, H., Li, G., Chen, B. 2013. L_1-Medial Skeleton of Point Cloud. ACM Trans. Graph. 32, 4, Article 65 (July 2013), 8 pages. DOI = 10.1145/2461912.2461913 http://doi.acm.org/10.1145/2461912.2461913.

DOI: http://doi.acm.org/10.1145/2461912.2461913

(a) A deer model. (b) Raw scan. (c) L_1-medial skeleton.

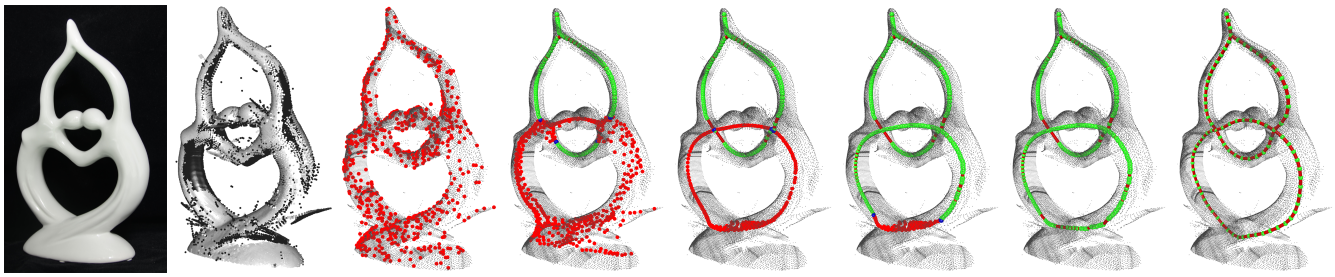
Figure 1: *Given an unorganized, unoriented, and incomplete raw scan with noise and outliers, our L_1-medial skeleton algorithm is able to extract a complete and quality curve skeleton.*

where the first term is a localized L_1-median of Q, the second term $R(X)$ regularizes the local point distribution of X, I indexes the set of projected points X, and J indexes the set of input points Q. The weight function $\theta(r) = e^{-r^2/(h/2)^2}$ is a fast decaying smooth function with support radius h defining the size of the supporting local neighborhood for L_1-medial skeleton construction.

The main advantage of our definition (1) is that it does not place strong requirements on the quality of the set of points representing the input shape. In particular, it can be directly applied to a raw point cloud acquired by a range scanner and the use of L_1-medians leads to robustness against various imperfections of the data. This is in contrast to the majority of existing methods for curve skeleton computation, which require the input shape to be complete, watertight or represented by fine tessellations [Chung et al. 2000; Dey and Sun 2006; Au et al. 2008; Hassouna and Farag 2009; Tagliasacchi et al. 2012; Willcocks and Li 2012].

Extracting a skeletal representation from a 3D shape is known to be an effective means for shape abstraction and consequently an effective tool for shape analysis and manipulation [Cornea et al. 2007]. Analyzing clean and well-represented shapes is important in its own right. However, there are also interests in directly processing raw point clouds which can be noisy, outlier-ridden, and even incomplete. An immediate application of curve skeletons extracted from a raw scan is surface reconstruction [Tagliasacchi et al. 2009]. In this context, it would be especially desirable if the skeletonization method does not rely on having accurate point normals since they are often unavailable or difficult to obtain from a raw point scan [Mullen et al. 2010]. Our L_1-medial skeletons have been designed to operate on raw point data directly, as shown in Figure 1.

Among the few works that have been proposed for curve skeleton extraction from point data, the method of Tagliasacchi et al. [2009] is the most notable and represents the state of the art. Their definition of a curve skeleton relies on a notion of generalized rotational symmetry axis (ROSA) and as such it assumes the input shape to be predominantly cylindrical. Correspondingly, the construction of each skeletal point requires a search for a 2D local neighborhood, namely a circular cross section. This highlights the key difference



(a) (b) (c) (d) (e) (f) (g) (h)

Figure 2: *Overview of L_1-medial skeleton extraction. Given an incomplete and noisy raw scan (b) of the object in (a), we randomly select a subset of samples, shown in red in (c). These points are iteratively projected onto a skeletal point cloud with a gradually increasing neighborhood size (d-g). After down-sampling, smoothing, and re-centering, the final curve skeleton is obtained (h).*

to our L_1-medial operator, which is inherently 3D without requiring the cylindrical shape prior. Not less importantly, our projection operator works directly on raw inputs, unlike ROSA which requires normal information as well as a separate preprocessing for noise and outlier removal. Experimental results show that our algorithm outperforms ROSA even when the latter is preceded by a state-of-the-art point cloud consolidation scheme [Huang et al. 2009]. Last and not the least, our algorithm is significantly faster than ROSA.

The main contributions and advantages of our method include:

- Curve skeleton extraction from general 3D shapes represented by point sets without prior assumptions on the shape geometry (e.g., cylindrical) or topology.
- Directly operating on raw scan data, that is, without preprocessing which may include denoising, outlier removal, normal estimation, spatial discretization, data completion, or (global or local) mesh reconstruction or parameterization.
- A simple formulation (1) that leads to fast and robust skeleton extraction based on default parameter settings.

2 Related work

Skeletal shape representations have been intensely studied in various fields and utilized in a variety of applications for shape modeling and analysis. The best known example is the medial axis transform (MAT) [Blum 1967], which belongs to the larger class of medial representations [Siddiqi and Pizer 2009]. While the MAT of a 2D shape is a 1D skeleton, for a 3D model, the MAT is generally composed of 2D surface sheets forming a non-manifold structure. In computer graphics, curve skeletons [Cornea et al. 2007] are more widely adopted due to their compactness and ease of manipulation, e.g., for character animation. In this section, we focus exclusively on related works developed for curve skeleton extraction.

Most previous methods for curve skeleton extraction have been designed to work on watertight surface meshes. Representative approaches include surface contraction via mean curvature flow [Au et al. 2008; Chuang and Kazhdan 2011; Tagliasacchi et al. 2012], coupled graph contraction and surface clustering [Jiang et al. 2013], and the use of distance transforms [Dey and Sun 2006], centroidal Voronoi tessellation [Lu et al. 2012], Reeb graph construction [Hilaga et al. 2001], or mesh segmentation [Katz and Tal 2003]. These approaches all depend on mesh connectivity or surface-based measures to control the skeletonization process.

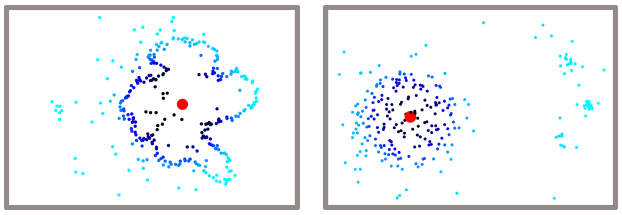
Sharf et al. [2007] generate a curve skeleton by tracing the front of a smoothly grown blob inside a volume defined by a point cloud. With significant missing data, it may be difficult to robustly detect the shape's interior and the blob may leak outside the shape. Li et al. [2010] also develop a deformable model called arterial snakes for curve abstraction from scanned shapes formed by tubular sections. They aim at extracting a set of functional curves that capture the structure of a shape, focusing on topology recovery. Cao et al. [2010] extend Laplacian-based contraction [Au et al. 2008] to point cloud skeletonization. However, the input point cloud needs to be sufficiently clean and dense to build a point set Laplacian. Recently, Livny et al. [2010] propose an algorithm to reconstruct skeletal structures from tree data. There have also been works [Bucksch et al. 2010; Natali et al. 2011] which rely on Reeb graph construction over point clouds to obtain a coarse characterization of the part structure of a shape.

The most related work is that of Tagliasacchi et al. [2009] on incomplete point clouds. Their formulation relies on a cylindrical shape prior and sufficiently accurate point normals to compensate for missing data. For raw inputs, noise and outliers need to be pre-filtered using a scheme such as locally optimal projection (LOP) [Lipman et al. 2007]. In contrast, our algorithm computes a medial curve skeleton by directly working on a noisy, outlier-ridden, and possibly incomplete raw scan without requiring reliable normal estimation or an assumption of cylindrical shape parts.

The concept of L_1-median has long been known in statistics [Weber 1909; Small 1990]. Recently, L_1-median has been successfully applied to point set processing [Lipman et al. 2007; Huang et al. 2009; Avron et al. 2010], all in the context of robust data fitting. An important novelty of our work lies in the utilization of L_1-medians for a new application, namely, curve skeleton extraction from raw point clouds. Without building any point connectivity or estimating point normals, we directly project point samples onto their local centers as L_1-medians with growing neighborhood and push the projected samples via conditional regularization to obtain a uniform distribution of samples along skeleton branches.

3 Overview

The input to our algorithm is an unorganized set of points $Q = \{q_j\}_{j \in J} \subset \mathbb{R}^3$, typically unoriented, unevenly distributed, and containing noise and outliers. The output is a curve skeleton representing a one-dimensional local center of the shape underlying the input Q. The main steps of the algorithm (Section 4) are as follows. A set of points (red in Figure 2(c)) is randomly sampled from the given raw scan. Each point is iteratively projected and redistributed to the center of the input points within its local neighborhood. The size of the neighborhood is gradually increased to handle



(a) (b)

Figure 3: *Insensitivity of the L_1-median (red dot) to noise and outliers in the data. The bluish shade of the points reflects the relative weight with respect to the L_1-median center.*

structures of different levels of details, yielding a set of clean and well-connected skeletal points, e.g., as shown in Figures 2(d-g).

If the input point cloud is highly non-uniform, the skeletal points obtained by the above algorithm may be non-uniform as well; see Figure 7. In addition, for highly incomplete point clouds, the generated branches may be off-centered; see Figure 8. To alleviate these two problems, Sections 4.3 and 4.4 present two enhancements: density-based weighting and re-centering. The final, enhanced L_1-medial skeleton is shown in Figure 2(h).

4 L_1-medial skeleton and construction

Consider the problem of finding the location x of a point which has the minimum total Euclidean distance to a set of points, i.e.,

$$x = \operatorname{argmin} \sum_{j \in J} \|x - q_j\|.$$

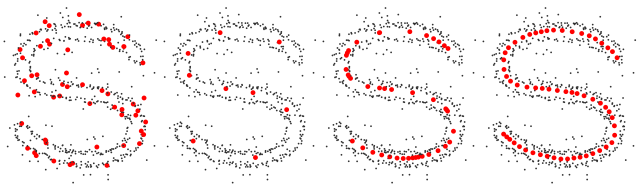
In statistics, the solution of this optimization problem is the spatial median or L_1-median, considered by Small [1990]. When the points $\{q_j\}_{j \in J}$ are not collinear, the L_1-median x is unique [Milasevic and Ducharme 1987].

The first term in our fundamental definition (1) can be viewed as a localized version of L_1-median. The localization is associated with the finite support radius h in the weight function θ. Different from other L_1-based approaches, e.g., [Lipman et al. 2007; Huang et al. 2009], whose goal is to form a clean point set representing a surface, the objective here is to compute a skeletal point cloud goving a 1D representation of the input geometry. In other words, we are seeking a *set* of *local L_1-median centers* of the underlying shape defined by Q. One key advantage of using L_1-medians is that, unlike the usual mean average, L_1-medians are insensitive to noise and outliers in the data, as shown in Figure 3.

In this section, we present our construction algorithm for L_1-medial skeletons. First, we define the regularization term R which prevents the formation of point clusters. Section 4.2 describes in detail the curve skeleton generation procedure. Finally, we present two enhancements to the core algorithm, aimed at dealing with non-uniform point density and the off-centering problem, respectively.

4.1 Conditional regularization

Applying the L_1-median alone tends to yield a sparse distribution, where local centers are likely accumulated into a set of point clusters; see Figure 4(b). To avoid such clustering and maintain the proper medial geometry representation, we need to prevent further accumulation once points are already contracted onto their local center positions. This is achieved using $R(X)$ in (1), which adds a



(a) Input. (b) $\mu = 0$. (c) $\mu = 0.1$. (d) $\mu = 0.35$.

Figure 4: *Skeletal point clouds generated with conditional repulsion defined by different strengths (μ).*

repulsion force whenever a skeleton branch is formed locally. Such a penalization is referred to as *conditional regularization*.

We adopt classical weighted PCA to detect the formation of skeleton branches. At each point x_i (a row vector), we compute the eigenvalues and eigenvectors of 3×3 weighted covariance matrix

$$C_i = \sum_{i' \in I \setminus \{i\}} \theta(\|x_i - x_{i'}\|)(x_i - x_{i'})^\top (x_i - x_{i'}),$$

where all eigenvalues $\lambda_i^0 \le \lambda_i^1 \le \lambda_i^2$ are real-valued and the corresponding eigenvectors $\{\mathbf{v}_i^0, \mathbf{v}_i^1, \mathbf{v}_i^2\}$ form an orthogonal frame, i.e., the principal components of the point set. Then we define the value

$$\sigma_i = \sigma(x_i) = \frac{\lambda_i^2}{\lambda_i^0 + \lambda_i^1 + \lambda_i^2} \tag{2}$$

as the *directionality degree* of x_i within a local neighborhood. The closer σ_i is to 1, the smaller λ_i^1 and λ_i^0 are compared to λ_i^2; and hence, the more points around x_i are aligned along a branch.

To conditionally apply the repulsion force, we define our regularization function as

$$R(X) = \sum_{i \in I} \gamma_i \sum_{i' \in I \setminus \{i\}} \frac{\theta(\|x_i - x_{i'}\|)}{\sigma_i \|x_i - x_{i'}\|},$$

where the $\{\gamma_i\}_{i \in I}$'s are balancing constants among X.

When the gradient of the energy in (1) equals to zero, the following relation is satisfied at every point location with fixed coefficients:

$$\sum_{j \in J} (x_i - q_j)\alpha_{ij} - \gamma_i \sum_{i' \in I \setminus \{i\}} \frac{x_i - x_{i'}}{\sigma_i} \beta_{ii'} = 0, \ i \in I,$$

where $\alpha_{ij} = \frac{\theta(\|x_i - q_j\|)}{\|x_i - q_j\|}$, $j \in J$; $\beta_{ii'} = \frac{\theta(\|x_i - x_{i'}\|)}{\|x_i - x_{i'}\|^2}$, $i' \in I \setminus \{i\}$. After rearranging and setting

$$\mu = \frac{\gamma_i \sum_{i' \in I \setminus \{i\}} \beta_{ii'}}{\sigma_i \sum_{j \in J} \alpha_{ij}}, \ \forall i \in I,$$

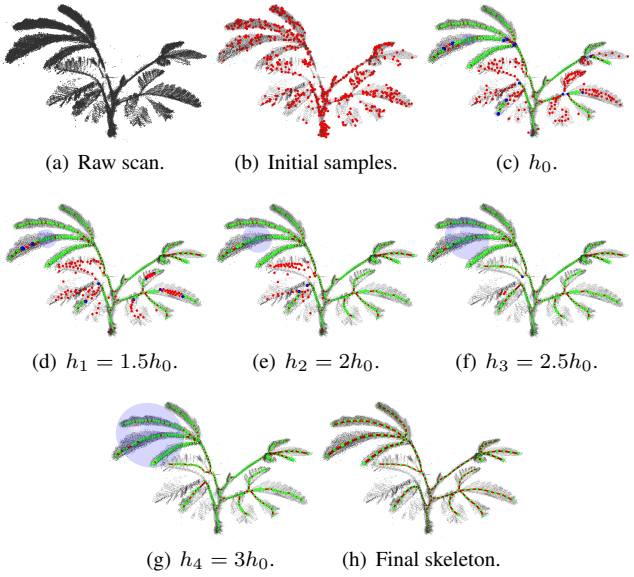
we get

$$(1 - \mu\sigma_i)x_i + \mu\sigma_i \frac{\sum_{i' \in I \setminus \{i\}} x_{i'} \beta_{ii'}}{\sum_{i' \in I \setminus \{i\}} \beta_{ii'}} = \frac{\sum_{j \in J} q_j \alpha_{ij}}{\sum_{j \in J} \alpha_{ij}}. \tag{3}$$

Note that (3) can be viewed as a system of equations with X as unknowns, i.e., $AX = BQ$. We therefore require $0 \le \mu\sigma_i < 1/2$ to ensure that A is strictly diagonally dominant and therefore non-singular. This yields the solution to the system: $X = A^{-1}BQ$.

Applying a fixed point iteration, where given the current iterate $X^k = \{x_i^k\}$, $k = 0, 1, \ldots$, we have the next iterate, $\forall i \in I$,

$$x_i^{k+1} = \frac{\sum_{j \in J} q_j \alpha_{ij}^k}{\sum_{j \in J} \alpha_{ij}^k} + \mu\sigma_i^k \frac{\sum_{i' \in I \setminus \{i\}} (x_i^k - x_{i'}^k)\beta_{ii'}^k}{\sum_{i' \in I \setminus \{i\}} \beta_{ii'}^k}, \tag{4}$$

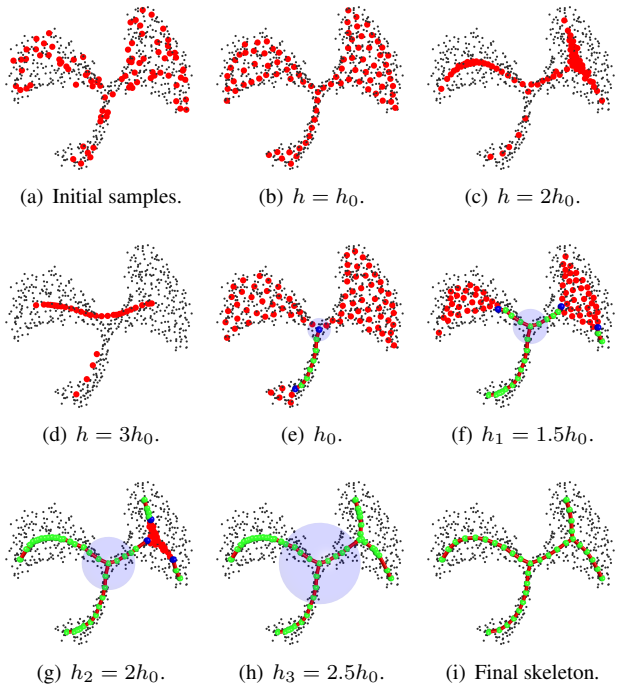


(a) Raw scan. (b) Initial samples. (c) h_0.

(d) $h_1 = 1.5h_0$. (e) $h_2 = 2h_0$. (f) $h_3 = 2.5h_0$.

(g) $h_4 = 3h_0$. (h) Final skeleton.

Figure 5: *Iterative point contraction and branch identification and growth by gradually increasing the neighborhood size (h_0 to h_4 indicated with a transparent blue ball). We select 5% of the input points as initial samples (b). In each intermediate step, branch points are labeled in green, non-branch points in red, and bridge points (points from which new branches grow) are in blue. Final skeleton (h) is formed after down-sampling and smoothing.*



(a) Initial samples. (b) $h = h_0$. (c) $h = 2h_0$.

(d) $h = 3h_0$. (e) h_0. (f) $h_1 = 1.5h_0$.

(g) $h_2 = 2h_0$. (h) $h_3 = 2.5h_0$. (i) Final skeleton.

Figure 6: *Comparing skeletal point clouds generated using fixed-size neighborhoods (three choices of size h shown in (b-d)) and result obtained (h) via iteration over growing neighborhood size (e-h). Color choices and sample selection are the same as in Figure 5.*

where $\alpha_{ij}^k = \frac{\theta(\|x_i^k - q_j\|)}{\|x_i^k - q_j\|}$, $j \in J$; $\beta_{ii'}^k = \frac{\theta(\|x_i^k - x_{i'}^k\|)}{\|x_i^k - x_{i'}^k\|^2}$, $i' \in I \backslash \{i\}$; $\sigma_i^k = \sigma(x_i^k)$. As adaptive $\sigma_i^k \in (0, 1]$ can be computed automatically to adjust the local repulsion force along the points aligning direction, we shall only select the parameter μ from $[0, 1/2)$ to control the global level of penalty applied on accumulated points; see Figure 4. We empirically use $\mu = 0.35$ as the default setting for all final results presented in this paper.

4.2 Curve skeleton via iterative contraction

Given a neighborhood size h, the aforementioned iterative projection scheme (4) produces a set of points $X = \{x_i\}_{i \in I}$, which represents the L_1-medians of the local neighborhoods. In simple cases, these points form a skeleton of the underlying shape directly; see Figure 4(d). However, for more complex shapes, only some of these points represent branches of the skeleton, whereas others require further contraction; see Figures 5(c) and 6(e).

In what follows, we first describe how we identify and label points belonging to skeleton branches (called *branch points* and colored green in figures). Then we explain how to select *bridge points* (colored in blue) from the remaining *non-branch points* (colored in red) and use them to maintain the connectivity between the two groups. Finally we show how the branches grow and merge iteratively by gradually enlarging the neighborhood size.

We start from an initial set of sample points, which are all considered as non-branch points. The sample points are contracted based on an initial neighborhood size $h_0 = 2d_{bb}/\sqrt[3]{|J|}$ by default, where d_{bb} is the diagonal length of the input Q's bounding box and $|J|$ is the number of points in Q. To determine whether to label a point as branch point after contraction, we employ the same directionality degree measure σ (2) used in Section 4.1. Specifically, we first compute σ_i for all non-branch points x_i and smooth them within the respective K-nearest neighborhood ($K = 5$ by default) to remove isolated outliers, i.e., $\sigma_i = \Sigma_{j \in Knn(i)} \sigma_j / K$. If, after smoothing, $\sigma_i > 0.9$, then x_i is considered as a candidate for branch point since points in x_i's neighborhood are well-aligned skeleton-wise.

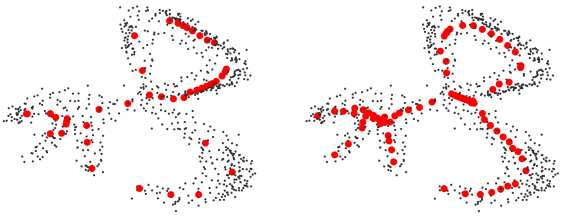
To identify branch points from these candidates, we first locate a seed point, namely x_0 that has the largest σ value, and trace from it to nearby candidates along the dominant PCA direction while requiring

$$\cos(\angle(x_i\overrightarrow{x_{i-1}}, x_i\overrightarrow{x_{i+1}})) \leq -0.9, \; i = \ldots, -1, 0, 1, \ldots$$

Tracing stops when there is no candidate in the local neighborhood satisfying the above criterion. If at least five points (by default) are found during the trace, these points are labeled as branch points. Otherwise, they are removed from the candidates. This procedure repeats from a new seed, one having the largest σ among the remaining candidates, until all candidates are processed.

As shown in Figures 5(c) and 6(e), the procedure described above leaves some sample points labeled as non-branch points; they require further contraction under a larger neighborhood size h to form new branches. However, relying on a *fixed* h, large or small, that remains unchanged throughout the contraction process, would not work. Figure 6 provides an example where with a large h, points representing small subparts are "over-contracted" so as to not maintain the medial structure. Our solution is to *gradually* increase h while excluding already identified branch points from further contraction. That is, $h_i = h_{i-1} + \Delta h, \; i = 1, 2, \ldots$ with $\Delta h = h_0/2$ by default, until all points become branch points.

Projecting the remaining non-branch points while fixing the branch points may separate the two groups, resulting disconnected skeleton branches. To address this problem, we select a bridge point along



(a) Without density weights. (b) With density weights.

Figure 7: *Under non-uniform point distribution, skeleton points tend to move close to high-density regions (a). Density weights allow the points to be better distributed and centered (b).*

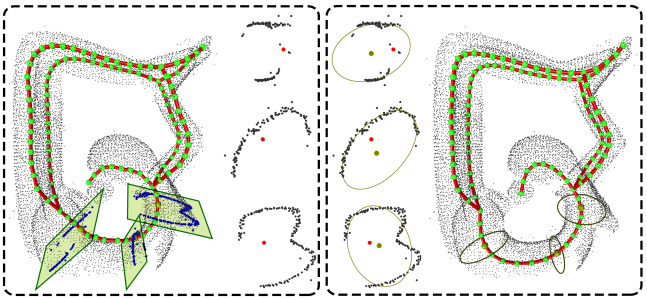


Figure 8: *Skeleton re-centering (red point to green point in the ellipses) on an incomplete raw scan via local ellipse fitting.*

each end of an identified skeleton branch. Denoting the endpoint of the branch as e, then its corresponding bridge point b is a non-branch point, satisfying i) along the direction of the branch, i.e., the angle between $\overrightarrow{eb}$ and the branch direction is smaller than 90^o; ii) closer to e than any other non-branch points that satisfy i); and iii) the distance $\|b - e\|$ is small.

These bridge points provide proper connectivity between branch points and non-branch points: each bridge point is linked to its corresponding endpoint of a branch and hence remains connected with the existing branch. On the other hand, as a non-branch point, it participates in further contraction and therefore is part of the new branches to be formed. As a result, branches found under different neighborhood sizes are connected, forming a complete skeleton.

Two additional rules are applied during contraction: i) when a local neighborhood contains two or more bridge points, they are collapsed into one branch point, which is connected to all branches to which the original bridge points belong; ii) non-branch points are removed when they are close to an existing branch but not along its aligning direction or when there are no other points in their neighborhood. The first rule leads to effective creation of joints, whereas the second helps clean up isolated points.

Once all points are contracted onto the skeleton, down-sampling and smoothing are applied to each branch independently to reduce redundancy. Figures 5(h) and 6(i) show the final L_1-medial skeletons generated for a 3D and 2D point cloud, respectively.

4.3 Density-based weighting

While L_1-medians are robust to outliers as shown in Figure 3, if the given point cloud is highly non-uniform, the local center tends to be biased towards regions with higher point density; see Figure 7(a). To alleviate this, we propose to incorporate locally adaptive density weights into the iterative scheme (4).

Let us define the weighted local density for each point q_j in the original input Q as

$$d_j = 1 + \sum_{j' \in J \setminus \{j\}} \theta(\|p_j - p_{j'}\|),$$

with $\theta(\|p_j - p_{j'}\|) = e^{-\|p_j - p_{j'}\|^2/(h_d/2)^2}$, and embed it into (4). Then the projection for point x_i^{k+1} becomes

$$x_i^{k+1} = \frac{\sum_{j \in J} q_j \alpha_{ij}^k / d_j}{\sum_{j \in J} \alpha_{ij}^k / d_j} + \mu \sigma_i^k \frac{\sum_{i' \in I \setminus \{i\}} (x_i^k - x_{i'}^k) \beta_{ii'}^k}{\sum_{i' \in I \setminus \{i\}} \beta_{ii'}^k}.$$

Thus, the influence of dense point regions in the set Q is relaxed by the weighted local densities in the first term; compare the resulting skeletal point clouds in Figures 7(a) and 7(b). Note that the locality of point density is controlled by the supporting neighborhood parameter h_d in the function $\theta(\|p_j - p_{j'}\|)$. We set $h_d = h_0/2$ by default, where h_0 is the initial smallest radius we use to apply L_1-median projection with conditional regularization.

4.4 Re-centering

By design, the generated L_1-medial skeleton goes through the median location of nearby points from the input point cloud. However, if large portions of the point cloud are missing (left in Figure 8), an additional re-centering step can be applied to fine-tune the location of the curve skeleton. To maintain efficiency and avoid distortion, we run re-centering separately on each branch after down-sampling, and then the curved branch is smoothed (right in Figure 8).

Taking one branch as example, for each point x on the branch, we define the plane P_x that passes through x and has a normal aligned with the direction of the branch at location x. Next, points from the input point cloud that are close to P_x are projected over it. For most natural shapes, we may expect the projected points to approximate an ellipse. Hence, adjusting the location of x to the center of the ellipse c_x allows the skeleton curve to better represent the input point cloud; see Figure 8. Here the location of c_x is estimated by solving a nonlinear least squares problem [Gander et al. 1994], where the geometric fitting error is minimized using Gauss-Newton.

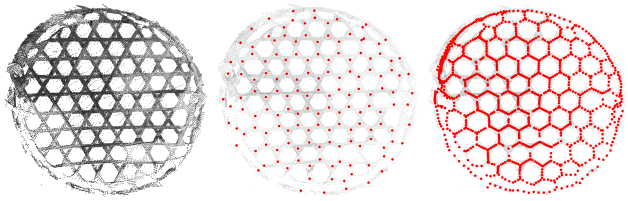
5 Results and discussions

We tested our skeleton extraction algorithm on a variety of raw scans. The 3D models chosen possess different geometric and topological characteristics: plants (Figure 5) along with other models that exhibit heavy branching structures (Figures 1, 11 and 13), curve networks (Figure 9), high-genus objects (Figures 2, 13, 9 and 14), and shapes with non-cylindrical geometry (Figures 9, 10 and 11). All the input point clouds to our algorithm are unorganized and unoriented raw scans without any preprocessing for noise or outlier removal. These inputs exhibit varying degrees of data artifacts including noise, non-uniform acquisition density, and missing data.

The default parameter sets in Table 1 are applied throughout all the presented experiments except for the example in Figure 4 and the coral example in Figure 13, where we demonstrate the influence of parameter change. The average running time for our algorithm on a

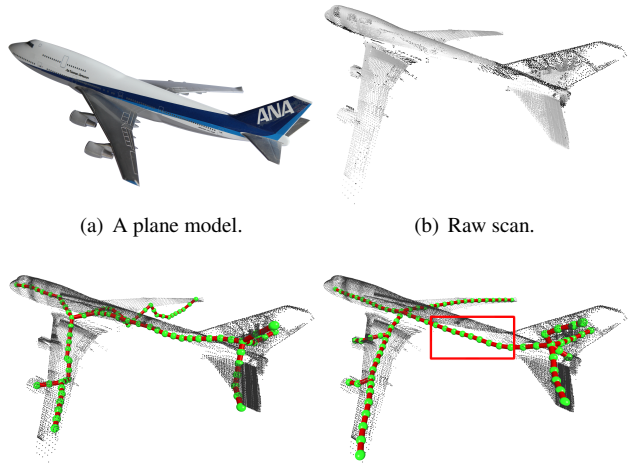
Parameter	μ	h_0	Δh	h_d	K
Default value	0.35	$2d_{bb}/\sqrt[3]{\|J\|}$	$h_0/2$	$h_0/2$	5

Table 1: *The default settings used for user controllable parameters.*



(a) Raw scan. (b) w/o regularization. (c) w/ regularization.

Figure 9: *The skeletal point clouds generated for a bamboo basket. Without conditional regularization (b), points are accumulated into a few clusters. Conditional regularization can effectively push points apart along the skeletal direction (c).*



(a) A plane model. (b) Raw scan.

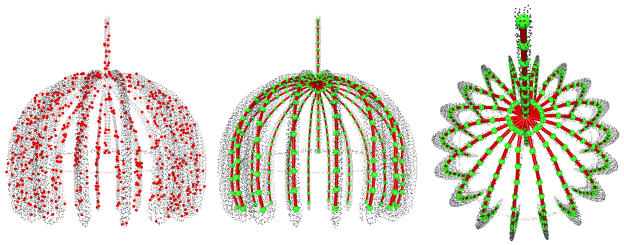
(c) ROSA skeleton. (d) L_1-medial skeleton.

Figure 10: *L_1-medial skeleton constructed (d) correctly captures the geometry of the non-cylindrical parts of the plane model (the wings and the tail). Note also that in the highlighted area, the skeleton is correctly placed outside the envelop of the input points through re-centering. In contrast, ROSA skeleton (c) with its re-centering step [Tagliasacchi et al. 2009] still yields unnatural results since the input violates the approximate cylindrical prior.*

scan with 100K points is about 1 minute on an Intel Core i7-2700K CPU @3.50GHz with 8GB RAM. In contrast, ROSA takes up to 5 minutes to process a point cloud of size 10K.

Results. The presented results justify various design choices in our algorithm and demonstrate its ability to handle the various data artifacts. In particular, Figures 1 and 2 demonstrate robustness against noise, outliers, point non-uniformity, and missing data. Figure 5 justifies the need for iterative growth of neighborhood size. The necessity of adding conditional regularization is shown in Figure 9. Figures 5, 9, and 10 demonstrate the handling of non-cylindrical shapes, whereas Figure 11 highlights our strength in dealing with complex topology. The dinosaur model in Figure 12 consists of shape parts whose sizes and shapes vary greatly. Yet, our algorithm with the default parameter setting produces a proper skeleton that can be immediately used for shape articulation.

Using default parameters for the coral model, on the other hand, does not yield a satisfactory skeleton, since the default initial neighborhood size h_0 is too large to capture the coral's fine structures and small holes; see Figure 13(b). Lowering h_0 and slowing down the

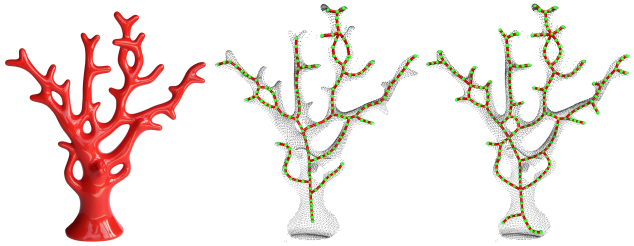


(a) Input scan. (b) Front view. (c) Top view.

Figure 11: *L_1-medial skeletons extracted from a raw lampshade scan. A joint connecting 20 branches is correctly identified.*

(a) A dinosaur model. (b) Raw scan. (c) Skeleton.

Figure 12: *L_1-medial skeleton (c) of a dinosaur model (a-b). The skeleton properly captures topological relations between different body parts, even though the parts differ greatly in size and shape.*



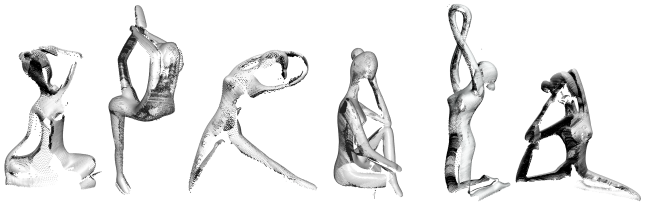
(a) A coral model. (b) Default h_0 & Δh. (c) Smaller h_0 & Δh.

Figure 13: *Curve skeletons extracted for a coral model (a) under different parameter settings. The one (b) obtained using the default parameters misses some fine structures and ignores three small holes. Using a smaller initial neighborhood size ($h_0 = d_{bb}/\sqrt[3]{|J|}$) and a slower growth rate ($\Delta h = h_0/4$) allows the algorithm to capture the complicated topology of the model (c).*

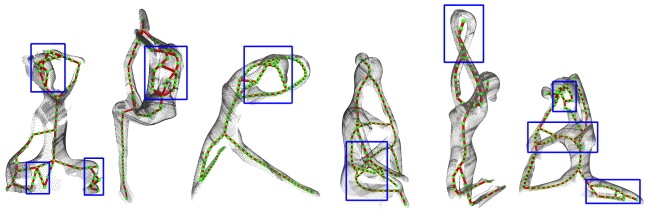
growth rate of the neighborhood size can effectively improve the result as shown in Figure 13(c), correctly capturing all the loops and coral twigs at the expense of a longer computation time. Note that at this level of details, the bottom of the coral is treated as two bumps, resulting in two branches that may look unnatural. This is a trade-off controlled by h_0 and Δh.

Comparison to ROSA. Figure 10 compares ROSA with our algorithm on the handling of non-cylindrical shapes. It is evident that our approach produces skeletons which better capture the more general geometry. Figure 14 makes the comparisons on a small set of yoga poses. Since the data were captured from few views, the amount of missing data is significant. Comparing with the ROSA skeletons shown, which represent the best results we were able to obtain following the authors' guidelines, the L_1-medial skeletons

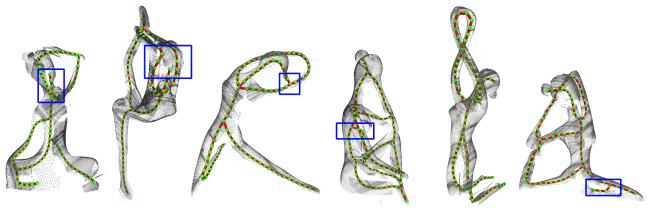
(a) Original models.



(b) Raw scans.



(c) ROSA skeletons.



(d) L_1-medial skeletons.

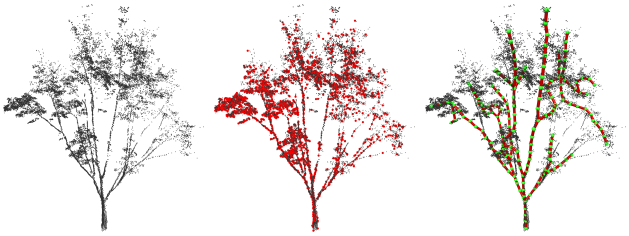
Figure 14: *Comparing ROSA skeletons (c) from [Tagliasacchi et al. 2009] with our L_1-medial skeletons (d) extracted from a set of raw scans (b). Blue boxes emphasize where the errors (small or large) occur. Note that ROSA requires correct normals on each input.*

contain fewer geometrical and topological errors. We stress again that all the ROSA results were obtained after the input scans were preprocessed for noise and outlier removal, followed by normal estimation [Huang et al. 2009]. In contrast, our approach does not need normal information nor separate data consolidation.

Relation to mean shift. Mean shift, which was originally presented in [Fukunaga and Hostetler 1975], is a simple iterative procedure that shifts each data point to the weighted average of data points in its neighborhood. Recall the first term in (4)

$$x_i^{k+1} = \frac{\sum_{j \in J} q_j \alpha_{ij}^k}{\sum_{j \in J} \alpha_{ij}^k},$$

which can be viewed as a mean shift to repeatedly move data points to the sample means, where $\alpha_{ij}^k(x_i^k - q_j)$ acts as a kernel function. Thus, our formulation (4) can be considered as a regularized mean shift that can generate skeletal points, whereas standard mean shift would move points into a few clusters like standard L_1-median.



(a) Raw scan. (b) Initial samples. (c) Skeleton.

Figure 15: *L_1-medial skeleton extraction from a tree scan (a). With the input points too sparse, our algorithm misses several branches and incorrectly runs the skeleton through the leaves in (c).*

Curve skeleton properties. Recalling the list of desirable properties for curve skeletons from the survey of Cornea et al. [2007], we would identify our approach as one which produces smooth, thin, connected, and centered skeletons with robustness and efficiency. Smoothness, thinness, and connectivity are by design. Centering is achieved via ellipsoidal fitting. Robustness and efficiency have been demonstrated experimentally. In most cases, the skeletons produced are homotopic to the underlying shape. However, claims on centeredness, robustness, or homotopy equivalence cannot be made as data quality continues to degrade. At last, the curve skeleton representation produced by our algorithm does not possess the reconstruction property. It is not likely that the regularization or contraction procedure is invertible.

Limitations. Like surface reconstruction, skeletonization from point clouds is an ill-posed problem in general, in particular, when there is missing data. If the amount of noise or missing data is too large, our algorithm may miss certain fine-scale structures and produce erroneous outputs, e.g., see Figure 15. Also, in Figures 5(h) and 11(b), we can notice that a few branches and a horizontal ring were missing. One possible way to partially alleviate the problem is to rely on a more sophisticated sampling scheme than the random sampling we apply now. For complex shapes that contain close-by surface sheets, it is difficult to distinguish points from different surfaces using position information only. As a result, our algorithm may produce skeletons with incorrect topology as highlighted in Figure 14(d). If more information is available, e.g., normals or outward directions directly derived from scanners, our algorithm can be enhanced to generate more accurate results.

6 Conclusion and future work

This paper presents a simple yet powerful approach for extracting curve skeletons from unorganized, unoriented, and incomplete 3D raw point clouds. It opens opportunities for analyzing various other imperfect representations, e.g., polygon soups, which may result from part composition or other editing operations. Applying high-level analysis over imperfect representations may lead to better reconstruction and better means to fix these representations.

As for future work, we would like to extend our technique to analyze imperfect (4D) spatio-temporal scans. It is also interesting to explore the ability of L_1-medians to capture meso-skeletons of models, such as a cup or baseball cap [Tagliasacchi et al. 2012], for which a centered one-dimensional solution becomes inappropriate. We would also like to explore the possibility of using the extracted skeletons to guide animation and/or deformation of point clouds directly without going through an explicit mesh reconstruction.

Acknowledgments

The authors would like to thank all the reviewers for their valuable comments. The raw scan data shown in Figure 8 is courtesy of Andrea Tagliasacchi. This work is supported in part by grants from NSFC (61103166, 61232011 and 61025012), Guangdong Science and Technology Program (2011B050200007), Shenzhen Innovation Program (CXB201104220029A and ZD201111080115A), Shenzhen Nanshan Program (KC2012JSJS0019A), Natural Science and Engineering Research Council of Canada (293127 and 611370) and the Israel Science Foundation.

References

AU, O. K.-C., TAI, C.-L., CHU, H.-K., COHEN-OR, D., AND LEE, T.-Y. 2008. Skeleton extraction by mesh contraction. *ACM Trans. on Graph (Proc. of SIGGRAPH) 27*, 3, 44:1–44:10.

AVRON, H., SHARF, A., GREIF, C., AND COHEN-OR, D. 2010. ℓ_1-sparse reconstruction of sharp point set surfaces. *ACM Trans. on Graph 29*, 5, 135:1–135:12.

BLUM, H. 1967. A transformation for extracting new descriptors of shape. *Models for the Perception of Speech and Visual Form*, 362–380.

BUCKSCH, A., LINDENBERGH, R., AND MENENTI, M. 2010. Skeltre: Robust skeleton extraction from imperfect point clouds. *The Visual Computer 26*, 10, 1283–1300.

CAO, J., TAGLIASACCHI, A., OLSON, M., ZHANG, H., AND SU, Z. 2010. Point cloud skeletons via laplacian based contraction. *Proc. IEEE Int. Conf. on Shape Modeling and Applications*, 187–197.

CHUANG, M., AND KAZHDAN, M. 2011. Fast mean-curvature flow via finite-elements tracking. *Computer Graphics Forum (Proc. of Eurographics) 30*, 6, 1750–1760.

CHUNG, J.-H., TSAI, C.-H., AND KO, M.-C. 2000. Skeletonization of three-dimensional object using generalized potential field. *IEEE Trans. Pat. Ana. & Mach. Int. 22*, 11, 1241–1251.

CORNEA, N. D., SILVER, D., AND MIN, P. 2007. Curve-skeleton properties, applications, and algorithms. *IEEE Trans. Vis. & Comp. Graphics 13*, 3, 530–548.

DASZYKOWSKI, M., KACZMAREK, K., HEYDEN, Y. V., AND WALCZAK, B. 2007. Robust statistics in data analysis - a review: Basic concept. *Chemometrics and Intelligent Lab. Sys. 85*, 2, 203–219.

DEY, T. K., AND SUN, J. 2006. Defining and computing curve-skeletons with medial geodesic function. *Symp. on Geom. Proc.*, 143–152.

FUKUNAGA, K., AND HOSTETLER, L. D. 1975. The estimation of the gradient of a density function with applications in pattern recognition. *IEEE Trans. Info. Theo. 21*, 32–40.

GANDER, W., GOLUB, G. H., AND STREBEL, R. 1994. Least squares fitting of circles and ellipses. *BIT Numerical Mathematics 34*, 558–578.

HASSOUNA, M. S., AND FARAG, A. A. 2009. Variational curve skeletons using gradient vector flow. *IEEE Trans. Pat. Ana. & Mach. Int. 31*, 12, 2257–2274.

HILAGA, M., SHINAGAWA, Y., KOHMURA, T., AND KUNII, T. L. 2001. Topology matching for fully automatic similarity estimation of 3D shapes. *Proc. of SIGGRAPH*, 203–212.

HUANG, H., LI, D., ZHANG, H., ASCHER, U., AND COHEN-OR, D. 2009. Consolidation of unorganized point clouds for surface reconstruction. *ACM Trans. on Graph (Proc. of SIGGRAPH Asia) 28*, 5, 176:1–176:7.

JIANG, W., XU, K., CHENG, Z., MARTIN, R. R., AND DANG, G. 2013. Curve skeleton extraction by coupled graph contraction and surface clustering. *Graphical Models 75*, 3, 137–148.

KATZ, S., AND TAL, A. 2003. Hierarchical mesh decomposition using fuzzy clustering and cuts. *ACM Trans. on Graph (Proc. of SIGGRAPH) 22*, 3, 954–961.

LI, G., LIU, L., ZHENG, H., AND MITRA, N. J. 2010. Analysis, reconstruction and manipulation using arterial snakes. *ACM Trans. on Graph (Proc. of SIGGRAPH Asia) 29*, 5, 152:1–152:10.

LIPMAN, Y., COHEN-OR, D., LEVIN, D., AND TAL-EZER, H. 2007. Parameterization-free projection for geometry reconstruction. *ACM Trans. on Graph (Proc. of SIGGRAPH) 26*, 3, 22:1–22:6.

LIVNY, Y., YAN, F., OLSON, M., CHEN, B., ZHANG, H., AND EL-SANA, J. 2010. Automatic reconstruction of tree skeletal structures from point clouds. *ACM Trans. on Graph (Proc. of SIGGRAPH Asia) 29*, 6, 151:1–151:8.

LU, L., LÉVY, B., AND WANG, W. 2012. Centroidal voronoi tessellations for line segments and graphs. *Computer Graphics Forum (Proc. of Eurographics) 31*, 2, 775–784.

MILASEVIC, P., AND DUCHARME, G. R. 1987. Uniqueness of the spatial median. *Ann. Statist. 15*, 1332–1333.

MULLEN, P., DE GOES, F., DESBRUN, M., COHEN-STEINER, D., AND ALLIEZ, P. 2010. Signing the unsigned: Robust surface reconstruction from raw pointsets. *Computer Graphics Forum 29*, 1733–1741.

NATALI, M., BIASOTTI, S., PATAN, G., AND BIANCAFALCIDIENO. 2011. Graph-based representations of point clouds. *Graphical Models 73*, 151–164.

SHARF, A., LEWINER, T., SHAMIR, A., AND KOBBELT, L. 2007. On-the-fly curve-skeleton computation for 3D shapes. *Computer Graphics Forum 26*, 323–328.

SIDDIQI, K., AND PIZER, S. 2009. *Medial Representations: Mathematics, Algorithms and Applications*. Springer.

SMALL, C. G. 1990. A survey of multidimensional medians. *International Statistical Review 58*, 3, 263–277.

TAGLIASACCHI, A., ZHANG, H., AND COHEN-OR, D. 2009. Curve skeleton extraction from incomplete point cloud. *ACM Trans. on Graph (Proc. of SIGGRAPH) 28*, 3, 71:1–71:9.

TAGLIASACCHI, A., ALHASHIM, I., OLSON, M., AND ZHANG, H. 2012. Mean curvature skeletons. *Computer Graphics Forum (Proc. of Symposium on Geometry Processing) 31*, 5, 1735–1744.

WEBER, A. 1909. *Über den Standort der Industrie*. Tübingen, J.C.B. Mohr (Paul Siebeck). English translation by C. J. Freidrich (1929): Alfred WebersTheory of the Location of Industries.

WILLCOCKS, C. G., AND LI, F. W. B. 2012. Feature-varying skeletonization - intuitive control over the target feature size and output skeleton topology. *The Visual Computer 28*, 6-8, 775–785.