

Distance field guided L_1 -median skeleton extraction

Chengfang Song¹ · Zhiqiang Pang¹ · Xiaoyuan Jing¹ · Chunxia Xiao¹

Published online: 24 October 2016
© Springer-Verlag Berlin Heidelberg 2016

Abstract We introduce a distance field guided L_1 -median method to extract topologically clean 1D curve skeleton from the point cloud model. We first voxelize the input point cloud, and compute the distance field for the point cloud. Then with the distance field, we extract the initial skeleton of the model using a multi-scale parameter controlled thinning method. Finally, we incorporate the initial skeleton into the L_1 -median optimization, and develop a distance field guided L_1 -median to effectively extract the complete skeleton from the point cloud. Our method exhibits the advantages of both the distance field based skeleton extraction methods and the L_1 -median skeleton extraction methods. Our skeleton extraction system is robust and effective, and can be applied to the raw scanned point cloud data.

Keywords Skeleton extraction · Point cloud · Skeleton alignment · L_1 -median · Distance Transform

Electronic supplementary material The online version of this article (doi:[10.1007/s00371-016-1331-z](https://doi.org/10.1007/s00371-016-1331-z)) contains supplementary material, which is available to authorized users.

✉ Chunxia Xiao
cxxiao@whu.edu.cn

Chengfang Song
songchf@whu.edu.cn

Zhiqiang Pang
1035922601@qq.com

Xiaoyuan Jing
jingxy_2000@126.com

¹ State Key Lab of Software Engineering, Wuhan University
School of Computer, Wuhan University, Wuhan 430072,
China

1 Introduction

Skeletons capture the essential topology of the underlying shapes, and have been widely used in various applications, such as model segmentation, registration, animation, and retrieval. Many skeleton extraction methods have been proposed, including high dimensional medial representation [31] and 1D curve representation [10]. Recently, owing to the simplicity of topology and ease of manipulation, 1D curve skeleton extraction has become a popular research topic in computer graphics community, for both closed polygonal meshes [2] and incomplete point cloud [15,34]. To extract satisfactory skeletons from point cloud, however, much space has been left to be improved, among which robustness and accuracy are two key problems need to addressed. In this paper, we focus on 1D skeleton extraction from the point cloud.

For the point cloud models, especially for the raw data acquired via laser scanning, to effectively extract their skeletons is still a challenging problem for the following reasons. First, unlike closed polygonal meshes, there is no topology information between the sampled points, many geometry operator, such as the mesh-decomposition based method [18], requires mesh connectivity, cannot be extended directly to them. Second, for the scanned raw data, the point cloud usually exhibits missing regions, heavy noise and outliers, to infer the interior skeleton of input model is a non-trivial problem. Finally, as skeleton extraction is an ill-posed problem, to obtain high quality results, many existing algorithms require tedious model-specific parameters tuning, thus, uniform parameter setting is preferable for processing point cloud models.

In this paper, inspired by the distance field based skeleton extraction method [12] and the L_1 -median skeleton extraction method [15], we propose a distance field guided

L_1 -median method to extract 1D curve skeleton from the point cloud. Our method works in the following steps. We first voxelize the input point cloud, and obtain a voxelized representation for the point cloud. This type of representation is even effective for model with large missing regions. With the voxelization results, we compute the distance field for the point cloud, where each voxel has a nearest distance from the model boundary. Then, from the distance field, we extract the initial skeleton of the model using a multi-scale parameter controlled thinning method. Finally, we incorporate the initial skeleton into the L_1 -median optimization and develop a distance field guided L_1 -median to extract a complete skeleton from the point cloud. Figure 1 illustrates the overview of the proposed algorithms.

Our method has following advantages. Compared with the distance field based method [12], our method can provide cleaner results. As distance field based methods [12] usually generate spotted locally minimal/maximal distance, which results in unconnected and incorrect skeletons. Compared with the L_1 -median method [15], our method is more robust and efficient. With distance field as the guidance in the iterative skeleton generation, the sampled points move to the center of region along the distance increasing direction, avoiding zigzagging moving. Finally, with the distance field, we can specify more suitable neighborhood size in L_1 -median skeleton optimization to obtain desirable results, which significantly alleviates the problem of trial-and-error.

Our main contribution is that we develop a distance field guided L_1 -median skeleton extracting method, which integrates the advantages of both the distance field and the L_1 -median projection skeleton extraction method. Our algorithm can be applied to the raw scanned data, requires only minimal user interaction to extract topologically clean and accurate skeletons, that is, we only need to set appropriate parameters to provide satisfactory skeletons.

2 Related work

Many algorithms have been proposed for skeleton extraction from static models, we refer the reader to the survey by Cornea and Min [10] and the most recent work [16]. Several skeleton extraction methods from surface sequences also have been proposed, please refer to [27]. In this section, we only review the most related methods, such as the distance field based skeleton extraction methods, and 1D curve skeletons for meshes and point cloud models. For methods on medial axis and other higher dimensional medial representations, please refer to [31].

Skeleton extraction using distance field A variety of distance field methods have been developed for skeleton extraction

[4,5,12,13,37]. The distance field can be computed using the fast marching method [29]. Most of these algorithms perform the following three steps: (1) finding the ridge points which locally center within the object, (2) pruning the insignificant extreme points, and (3) reconnecting the remaining extreme points. The main advantage of these methods is that the distance field computation is usually fast, the disadvantage is that pruning the insignificant extreme points is a non-trivial task, which may produce unclear skeletons. In this paper, inspired by parameter controlled volume thinning method [12], we extract the skeleton as the initial value for the following skeleton extraction optimization.

Skeleton extraction for mesh model Various methods have been developed to extract skeletons from the watertight surface meshes. Important works include mesh decimation based method [21], segmentation based method [18], field based approach [8,25], geodesic distance based method [11], mean curvature flow based surface contraction [2,9,33], and method coupling graph contraction and surface clustering [17]. However, these methods require the mesh connectivity to obtain the skeletons. For example, Katz and Tal [18] applied graph cut on the mesh model to perform mesh decomposition, while graph cut requires mesh connectivity. Au et al. [2] extracted skeleton from the mesh model by shrinking the mesh with constrained Laplacian smoothing, and the smoothing operator also requires mesh connectivity.

Skeleton extraction for point cloud Both [6] and [26] constructed Reeb graph over point clouds to compute skeletons. Sharf et al. [30] applied deformable model, which involved multiple fronts inside the model, to capture the model's volumetric shape. Then they tracked the fronts' centers, merged and filtered the insignificant branches to obtain the final curve skeleton. Tagliasacchi et al. [34] proposed a ROSA (rotational symmetry axis) method to extract skeleton from incomplete point cloud. This method assumed the input shape to be cylindrical, furthermore, ROSA required normal information, while normal is difficult to estimate for sample points on the raw scanned data. Cao et al. [7] extended the mean curvature flow based skeleton extraction method [2] to the point cloud models. Livny et al. [24] extracted the tree skeletal structures from point cloud models. Li et al. [20] employed arterial snakes to extract skeleton from the incomplete point cloud, while this algorithm focused on topology recovery. Verroust et al. [35] computed the level sets of the distance map using neighborhood graph to extract curve skeleton of tabular shape such as blood vessels. Kustra et al. [19] computed refined skeleton from raw medial-surface point clouds. Skeletons can also be used to recover the intrinsic reflection symmetries of shapes, Zheng et al. [36] applied curve skeletons to perform intrinsic symmetrization for the input shape.

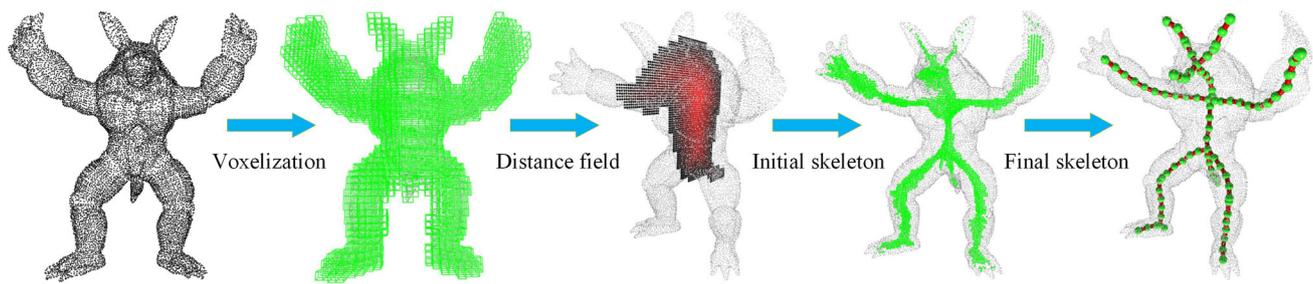


Fig. 1 System overview. The input cloud is first voxelized, then, we estimate the distance field. With the distance field, we compute the initial skeleton. Finally, we refine the initial skeleton using distance field guided L_1 median

More recently, Huang et al. [15] introduced a L_1 -medial projection operator to extract curve skeleton from 3D point cloud. As a state of the art method, without preprocessing, this algorithm can be directly operated on raw scanned data with poor quality to produce compelling results. However, in the iterative contraction procedure, the neighborhood size setting is important for obtaining satisfactory results. Although sophisticated adaptive neighborhood size tuning method have been given in [15], to process complex models, even using model-specific parameters tuning, to obtain desirable skeleton is a non-trivial task for this method [15].

3 System overview

The input to our method is one 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 1D curve skeleton $X = \{x_i\}_{i \in I} \subset \mathbb{R}^3$ representing a one-dimensional local center of the shape underlying the input Q . The main steps of the algorithm are as follows.

Distance field computing The input raw scan model is first uniformly voxelized using multi-scale dividing method. On the output voxelization, we apply fast marching method to compute the distance field for the model.

Initial skeleton extraction With the computed distance field, we produce an initial skeleton for the input model using the multi-scale parameter controlled thinning, which provides an initial skeleton used in the following skeleton optimization.

Skeleton refinement By incorporating the initial skeleton into the L_1 -median optimization, we develop a distance field guided L_1 -median to extract the complete skeleton from the point cloud.

In our system, the voxelization presentation with distance field provides the internal, external and boundary information of model, which guides the sample points to move to the center of the model. With the guidance of initial skeleton and the distance field, we improve the L_1 -medial method [15], and the projected samples will converge to the center of

model in one optimal way. Figure 1 gives the system overview of the proposed method.

4 Distance field computing

It is more difficult to voxelize the point cloud model than the closed mesh model. To achieve high-resolution voxelization for the point cloud, we first have to specify the boundary voxels of the point cloud model, then identify the outside voxels and the inner voxels. Finally, we refine these voxels. Similar to most of other methods, our method also cannot cope with the point clouds representing a non-orientable surface, such as the Klein bottle.

As point cloud is usually not closed, if the voxel is too small, the voxels cannot envelop the boundary of point cloud. To this end, we first build one bounding box for input model, and voxelize this bounding box with an initial voxel size, which is defined as $s_0 = d_{bb}/9$, (d_{bb} is the diagonal length of the bounding box). If a voxel contains one or more points of model, it is a boundary voxel. We specify an exterior voxel as seed and use flooding method to specify the outside voxels. With the boundary voxels and outside voxels, the remained voxels are inner voxels in the volume.

We then divide the volume with a multi-scale refinement strategy. In each refinement scale, we further uniformly divide each voxel into $3 \times 3 \times 3$ smaller sub-voxels. We do not change the status of the sub-voxels from the interior voxels and the exterior voxels. For each divided voxel from the boundary voxel, if it contains one or more sample points of the input model, it is labeled as boundary voxel; if one of its 26 neighbors is exterior voxel, it is labeled as exterior voxel; otherwise, it is labeled as interior voxel. We subdivide them in the same way progressively until the voxel resolution up to a given threshold $V^3 (V \times V \times V)$, (which is essentially adjustable) and V is set to 100 by default. Figure 2 shows the refinement procedure of voxelizing the dinosaur model.

For models with highly concave regions, mis-classification may occur for the sub-voxels of boundary voxels in these concave regions during the refinement procedure of voxelization.

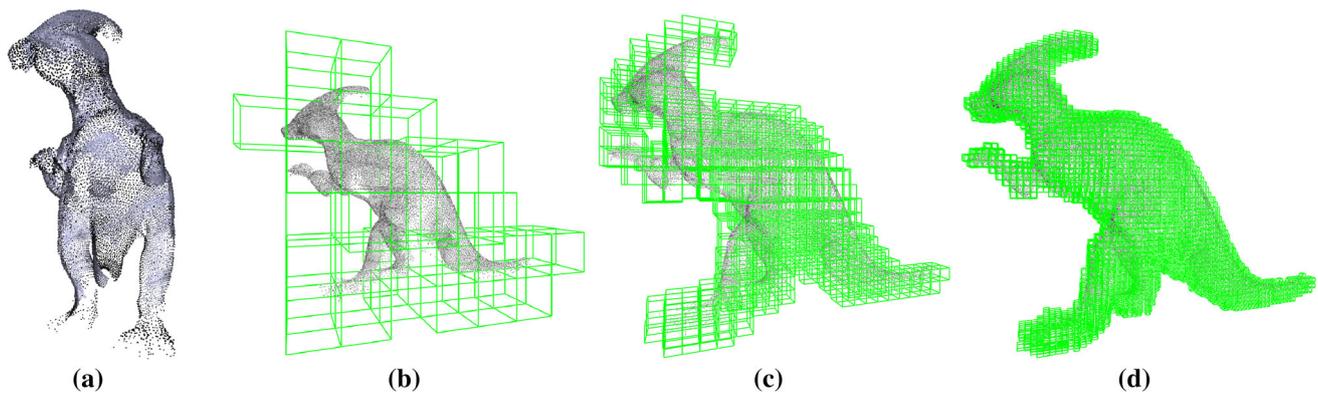


Fig. 2 Point cloud voxelization. **a** Input point cloud. **b–d** are voxelization results with different scales

It happens when an exterior voxel is surrounded by neighboring voxels containing points, it is mis-classified as interior. Although this rarely happens, this kind of mis-classification can be alleviated or even avoided by decreasing the (initial) voxel size of the multi-scale voxelization heuristically. Also note that even some voxels are mis-classified, they have no much impact on the final skeleton results, since we extract the skeletons using a global optimization system.

This scheme is effective even for models with noise, outliers, and missing areas. As illustrated in Fig. 2, we effectively voxelize the model components with large missing regions.

For raw inputs with heavy noise, outliers and missing regions, we can pre-filter the input model using the locally optimal projection (LOP) [23]. Actually, even if we pre-processed the input model with LOP, its topology would not be altered. Thus, we can obtain desirable skeleton that captures the essential topology of the model all the same.

For models with sparsely sampled regions, we resort to upsampling. Many point set upsampling methods have been proposed, such as moving least squares method (MLS) [1]. In our method, we employ a simple method. To perform upsampling, we first detect regions with insufficient sampling density according to the local sampling density ρ_i . We estimate ρ_i for each q_i by finding the sphere with minimum radius r_i centered at q_i that contains the k -nearest neighbors to q_i . Then ρ_i is defined as $\rho_i = k/(r_i)^3$. If ρ_i is below a given threshold value, then new sample points must be inserted. In our experiments, we insert the points using the simple linear interpolation, which works well for our skeleton extraction purpose. Figure 8 shows a lion model containing 8K points and its upsampled version containing 30K points after upsampling.

With the voxelization representation $P = \{p_i\}_{i \in L}$ of point cloud, where L is the total number of the voxels in P , we apply the fast marching method [29] to approximate the distance field for the voxelization representation. The distance field DT_p of each interior voxel p of a voxelized volume is the smallest distance from this voxel to the boundary of the

volume. Note that, to compute distance field for the model with many disconnected components, we can estimate the distance field for each component, respectively, to produce the initial skeletons. Figure 5 shows the color-coded distance fields of two models.

Note that for model with heavy noise/outliers, using our voxelization method, these noise/outliers will be contained in the voxels, and these voxels are classified into boundary voxels, as illustrated in Fig. 3c. In the distance field estimation using the fast marching method, these voxels are set the initial values (in our experiments, the initial value is zero). As these voxels are scattered and are disconnected with model, in the fast marching method, the distance field will not propagate in these voxels. We can remove these voxels from the generated distance field, and these voxels will not be included in the following skeleton extraction, as illustrated in Fig. 3d.

5 Initial skeleton extraction

Gagvani et al. [12] selected the ridge voxels having local maximal distance value as the candidates for curve skeleton points. They decided whether one voxel is a ridge voxel by performing the comparisons between the distance field value at a voxel and the average distance field value of its neighbors. That is, if a voxel meets the following condition, it is labeled as a skeleton point:

$$MNT_p < DT_p - TP$$

$$MNT_p = \sum_{i=1}^{26} DT_{p_i}/26, \quad p_i \in VP \quad (1)$$

where DT_p is the distance between voxel p and its nearest boundary voxel, VP is union of 26 neighboring voxels, $p_i \in VP$, TP is the thinning parameter, determining how close MNT_p should be to DT_p for p to be added to the skeleton. It controls the thickness of skeleton.

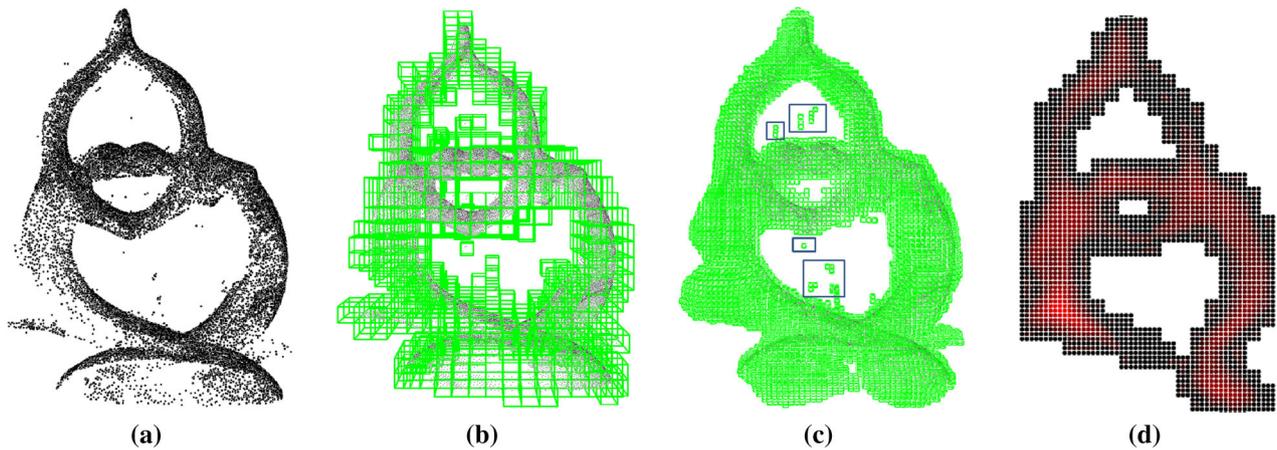


Fig. 3 Point cloud voxelization and distance field computing. **a** Input model, **b**, **c** are voxelization results with different scales, observing the voxelization results on the outliers, **b** is the color-coded slice of the distance field

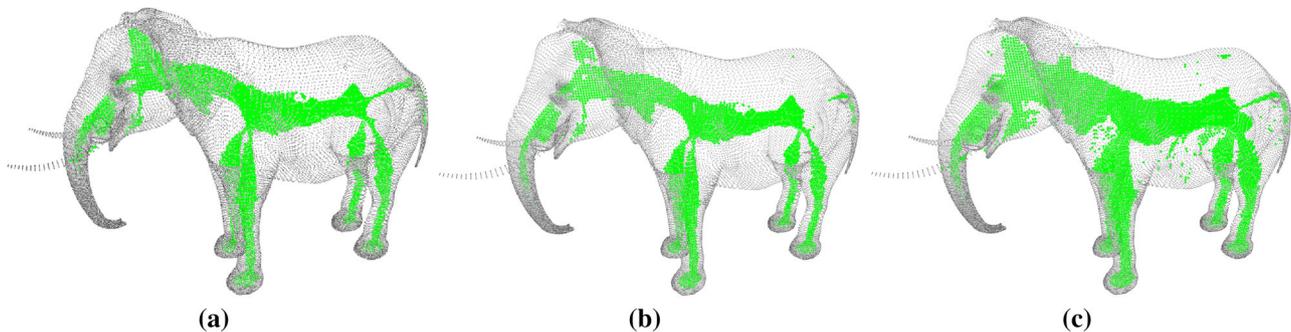


Fig. 4 Comparison of initial skeleton extraction methods. **a** Result of our method. **b** Result setting $TP = 0.7$. **c** Result setting $TP = 0.5$

While using this parameterized thinning method, users have to specify the value for TP , which is neither straightforward nor so easy to estimate. Inappropriate parameter TP setting will make the initial skeleton too thick or too thin, furthermore, some noise skeleton points will be introduced. To address this problem, we propose a multi-scale method to identify the initial skeleton points. We compute TP in three different scales (with neighborhood size of $3 \times 3 \times 3$, $5 \times 5 \times 5$ and $7 \times 7 \times 7$, respectively):

$$\begin{aligned} MNT_p^1 &= \sum_{i=1}^{26} DT_{p_i}/26, & MNT_p^1 &< DT_p - TP_1 \\ MNT_p^2 &= \sum_{i=1}^{124} DT_{p_i}/124, & MNT_p^2 &< DT_p - TP_2 \\ MNT_p^3 &= \sum_{i=1}^{342} DT_{p_i}/342, & MNT_p^3 &< DT_p - TP_3 \end{aligned} \quad (2)$$

For the 26 neighbors of each voxel p , we first compute their average distance field MNT_p^1 . Then TP_1 is defined as the difference between the average value of MNT_p^1 of all the voxels and DT_p of all the voxels in the model, that is, $TP_1 = c \cdot \frac{\sum_{p \in L} MNT_p^1 - \sum_{p \in L} DT_p}{|L|}$, and c is a coefficient between 0 and 1. We empirically set $c = 0.86$, and find it an appropriate

value for all of our experiments. TP_2 and TP_3 are evaluated in the similar way for the 124 neighbors and the 342 neighbors of each p , respectively. If voxel p meets all of the conditions of Eq.2, then we label p as the skeleton points.

As illustrated in Fig. 4, our method greatly improves the skeleton extraction results without tedious parameter setting. However, for complex model, as illustrated in Fig. 5c, the result is still not satisfactory, where there are many points with local minimal/maximal value using the approximate distance field. Thus, uniform parameter TP setting cannot handle all of situations, we need to refine the skeletons generated using distance field.

6 Skeleton refinement

We incorporate the initial skeleton into the L_1 -median optimization [15] to refine the skeleton results. Recently, L_1 -median [32] has been widely utilized in point cloud processing [3, 14, 15, 22, 23, 28]. Huang et al. [15] exploited local L_1 -median method to extract skeleton of the unoriented raw point scan.

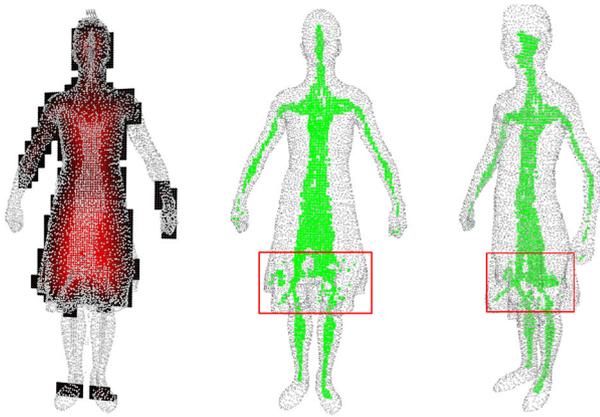


Fig. 5 *Left* Color-coded slice of the distance field of the model. *Middle* and *right* the extracted skeleton by the distance field (different view points)

Given an unoriented set of points $Q = \{q_j\}_{j \in J} \subset R^3$, the L_1 -medial skeleton can be obtained by using the optimal distribution of the projected points $X = \{x_i\}_{i \in I}$:

$$\begin{aligned} & \arg \min_X \sum_{i \in I} \sum_{j \in J} \|x_i - q_j\| \theta(\|x_i - q_j\|) \\ & + \sum_{i \in I} \gamma_i \sum_{i' \in I \setminus \{i\}} \frac{\theta(\|x_i - x_{i'}\|)}{\sigma_i \|x_i - x_{i'}\|} \end{aligned} \quad (3)$$

where the first term is a localized L_1 median of Q , the second term 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 . Parameter σ_i computed using weighted PCA is applied to detect the formation of the skeleton branches. $\{\gamma_i\}_{i \in I}$ are balancing constants among X .

In this method [15], appropriate support radius h setting and bridge points selection (bridge points are used to connect skeleton branches, and connect skeleton branch with other non-branch points) are vitally important to obtain good results. With appropriate parameters setting and bridge points selecting, this method can produce desirable skeleton for model such as Fig. 6a. However, to set appropriate parameters and select bridge points is a tedious task. As illustrated in Fig. 6b, for the same model with different pose, this method may not work well.

The L_1 -medial method tries to find the center of component as the skeleton points. As complex point cloud is usually composed of lots of disconnected components, if the neighborhood with radius h contains points coming from only one component of model, the sample point x_i may be moved to the center of the part. However, if the neighborhood contains several components of the model due to inappropriate radius h setting, it will be difficult for the sample point x_i

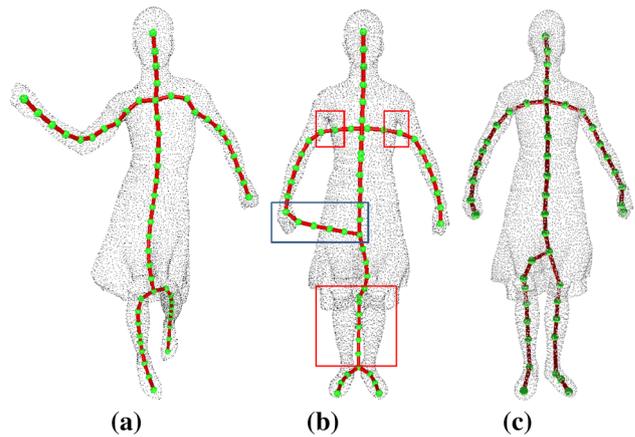


Fig. 6 *Left* appropriate support radius h setting produces good results. *Middle* wrong bridge points lead to uncorrected skeleton extraction result. *Right* using voxelization information to select the bridge points, we get much better result

to move to the center of right part, which will reduce to the wrong results. Furthermore, for model with several components with different sizes (each component also may have different size in its different parts), it is also difficult to set uniform h for all the components to produce high quality results. Thus, setting adaptive h is required to obtain good results. Although adaptive h setting techniques are given in [15], to produce satisfactory results is still tedious.

6.1 Distance field guided L_1 -medial

We apply the information of classified voxels and distance field to solve above problems. When iteratively projecting the sample points to the center of the components, we assure that they should move along the direction of distance field increasing. For bridge points selecting, the connection line between branch endpoint and bridge point must be always inside the model.

Initial skeleton produced in previous section is located at the central regions of the model components, and can provide useful information to guide the movement of the projected sample points. Based on this observation, we use initial skeleton to guide the convergence of sample points, and present the following distance field guided L_1 median samples projector:

$$\begin{aligned} & \arg \min_X \sum_{i \in I} \sum_{j \in J} \|x_i - q_j\| \theta(\|x_i - q_j\|) \\ & + \sum_{i \in I} \theta_i \left(\frac{n}{N} \right) \sum_{k \in K} \|x_i - s_k\| \theta(\|x_i - s_k\|) \\ & + \sum_{i \in I} \gamma_i \sum_{i' \in I \setminus \{i\}} \frac{\theta(\|x_i - x_{i'}\|)}{\sigma_i \|x_i - x_{i'}\|} \end{aligned} \quad (4)$$

where K is the index of initial skeleton points $S = \{s_k\}_{k \in K} \subset R^3$ and $\theta_i(r) = ne^{-r^2}$. For the neighborhood centering at x_i with radius h , N is the number of model points contained in the neighborhood, n is the number of initial skeleton points contained in the neighborhood.

The distance field guided L_1 median (Eq.4) indicates that initial skeleton has strong attraction to the projected sample points, and makes the projected points move to component center. When we set the gradient of the energy (Eq.4) to zero, we can obtain the following relation for each point location:

$$\sum_{j \in J} (x_i - q_j) \alpha_{ij} + \theta_i \sum_{k \in K} (x_i - s_k) \alpha_{ik} - \gamma_i \sum_{i' \in I \setminus \{i\}} \frac{x_i - x_{i'}}{\sigma_i} \beta_{ii'} = 0, \quad i \in I \tag{5}$$

where $\alpha_{ij} = \frac{\theta(\|x_i - q_j\|)}{\|x_i - q_j\|}$, $j \in J$; $\alpha_{ik} = \frac{\theta(\|x_i - s_k\|)}{\|x_i - s_k\|}$, $k \in K$; $\beta_{ii'} = \frac{\theta(\|x_i - x_{i'}\|)}{\|x_i - x_{i'}\|^2}$, $i' \in I \setminus \{i\}$.

We set

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

$$v = \frac{\theta_i \sum_{k \in K} \alpha_{ik}}{\sum_{j \in J} \alpha_{ij}}, \quad \forall i \in I \tag{6}$$

Then by rearranging Eq. (6) we get

$$(1 + v - u) x_i + u \frac{\sum_{i' \in I \setminus \{i\}} x_{i'} \beta_{ii'}}{\sum_{i' \in I \setminus \{i\}} \beta_{ii'}} = \frac{\sum_{j \in J} q_j \alpha_{ij} + \theta_i \sum_{k \in K} s_k \alpha_{ik}}{\sum_{j \in J} \alpha_{ij}} \tag{7}$$

Eq.7 can be considered as a system of equations with X as unknowns, i.e., $AX = BQ + SP$. As $v \geq 0$, and $u \geq 0$, if $0 \leq u < (1 + v)/2$, then matrix A is strictly diagonally dominant and is non-singular. The solution can be obtained by solving the system: $X = A^{-1}(BQ + SP)$.

In our implementation, similar to [15], we apply a fixed point iteration to solve above system. Given the current iteration $X^t = \{x_i^t\}$, $t = 0, 1, \dots$, the next iteration is computed as follows, $\forall i \in I$,

$$x_i^{t+1} = (u - v)x_i^t + \frac{\sum_{j \in J} q_j \alpha_{ij}^t + \theta_i \sum_{k \in K} s_k \alpha_{ik}^t}{\sum_{j \in J} \alpha_{ij}^t} - u \frac{\sum_{i' \in I \setminus \{i\}} x_{i'}^t \beta_{ii'}^t}{\sum_{i' \in I \setminus \{i\}} \beta_{ii'}^t} \tag{8}$$

where $\alpha_{ij}^t = \frac{\theta(\|x_i^t - q_j\|)}{\|x_i^t - q_j\|}$, $j \in J$; $\alpha_{ik}^t = \frac{\theta(\|x_i^t - s_k\|)}{\|x_i^t - s_k\|}$, $k \in K$; $\beta_{ii'}^t = \frac{\theta(\|x_i^t - x_{i'}^t\|)}{\|x_i^t - x_{i'}^t\|^2}$, $i' \in I \setminus \{i\}$; $\sigma_i^t = \sigma(x_i^t)$. As σ_i^t is defined as the directionality degree of x_i within a neighborhood, then $\sigma_i^t \in (0, 1]$. As $v \geq 0$ always holds, if we set from $[0, 1/2)$, matrix A is strictly diagonally dominant, and the sequence $x_i^0, x_i^1, x_i^2, \dots$, will converge to a position. In this paper, we set $u = 0.35$ for all the results.

In each iteration, the sample point x_i should moves along the direction with larger distance field. Let D_i^k be the distance value of point x_i^k at k th iteration. If in the next iteration, x_i^{k+1} moves to one voxel with less value: $D_i^{k+1} < D_i^k$, we keep x_i fixed in the iteration, that is, x_i does not move in this iteration.

Note that, if $0 \leq u < (1 + v)/2$, then matrix A is strictly diagonally dominant and is non-singular, using the fixed point iteration, the x_i^t will converge to a position. In addition, if we do not set above hard constraint, the moved sampled points, that is, the generated skeleton may penetrate out of the model. In our implementation, we found the iterations always converge and the sample points always move to the model center. It also should be pointed out, using above hard constraints, the iterations will converge relatively slow.

To set appropriate radius size h is important to obtain desirable results. In [15], Huang et al. set uniform value in the initial iterations. In the following contraction iterations, they increase h gradually to contract the non-branch points to the component center. In our method, the initial skeleton and the distance field provide useful cues to set adaptive radius h for the sample points. Let sample point x_i^t at the t th iteration has support radius h_i^t , and the neighborhood contains n initial skeleton points, $s_j, j \in n$. We compute the center of these initial skeleton points in the neighborhood as:

$$center = \sum_{j \in n} s_j \theta(\|x_i^t - s_j\|) / n \tag{9}$$

If the distance between $center$ and sample point x_i^t is smaller than the size of voxel, the next support radius h_i^{t+1} will be set as the distance field value of voxel which contains $center$. If the distance is larger than the size of one voxel, then $h_i^{t+1} = (1 + radio)h_i^t$, where $radio$ is the increasing radio of the support radius. Note that, in the initial iterations, we use the same support radius defined in [15], we find that it works well in our experiments.

Similar to [15], we use bridge points to produce a complete skeleton. With the distance field, we can select suitable bridge points to connect the skeleton branch to the non-branch points for updating the existing skeleton branch, or to create joints which connect the neighboring skeleton branches. When we select bridge point, if the connection curve of the branch endpoint e and the bridge point penetrates outside of the model, it is not a qualified bridge point. With the voxelization pre-

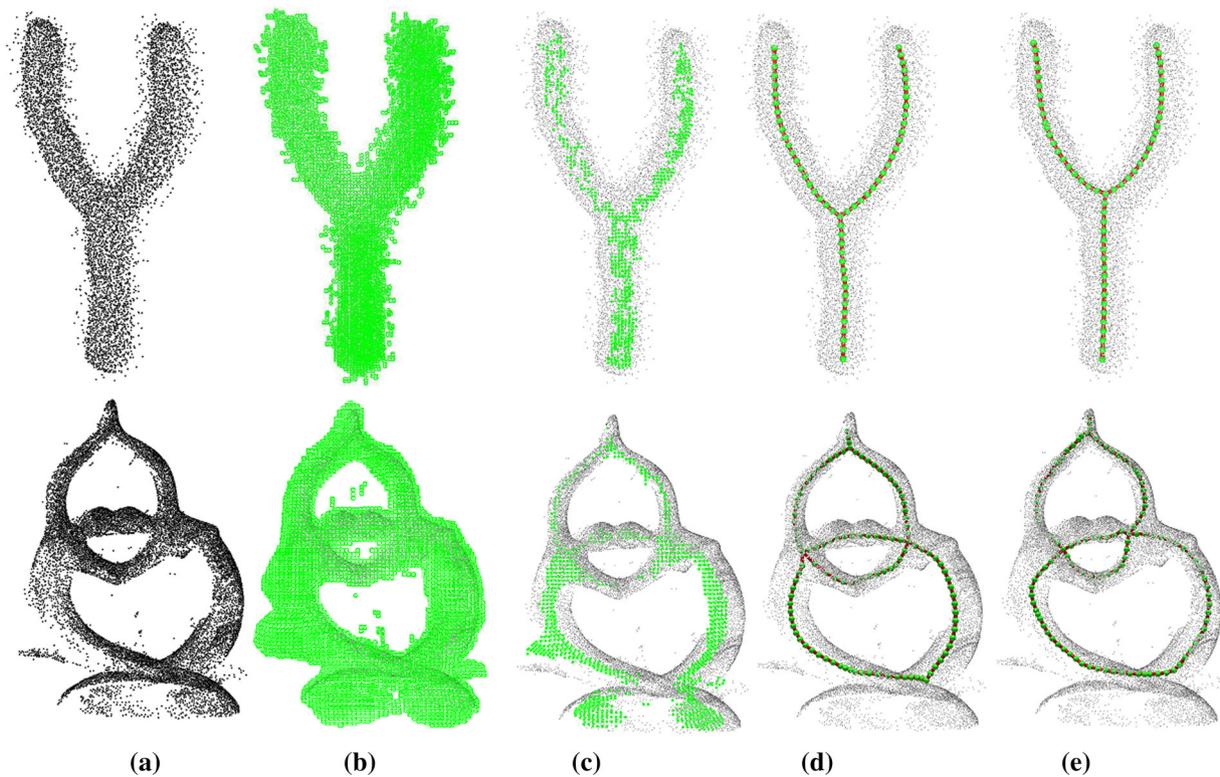


Fig. 7 Skeleton extraction comparisons. **a** Input models, **b** voxelization results, **c** initial skeletons, **d** our final skeletons, **e** results of [15]

sensation and voxel type information (boundary, outside and inner voxel), we can assure that skeleton will not penetrate outside of the model by selecting appropriate bridge point. That is, if connection curve passes through outside voxels, then this connection curve is not qualified, and we need to select new bridge points. In Fig. 6, we give the comparison results using and without using voxelization information for bridge point selection.

7 Results and discussion

We validate the effectiveness of the proposed methods by demonstrating the skeletons extracted from models of diverse shape and structure. We also compare our method with the most related method [15].

The running time of the proposed method is mainly consumed in the following four steps: voxelization, distance field computing, initial skeleton extraction, and final skeleton refinement. The former three steps are relatively fast, and the most time-consumption step is the latter. While accurate time consumption is model-specific, we evaluate the timing consumption of our method statistically. For a model with 35,000 points, if we take 1500 sample points in the projecting contraction, and voxelize the model with the resolution of $100 \times 100 \times 100$, it averagely takes 3 seconds for vox-

elization, 10 seconds for distance field computing, 3 seconds for initial skeleton extraction, and 24 seconds for skeleton refinement using L_1 median. By using the code presented by [15], it usually consumes about 32 seconds for the same model to produce the results. Our method costs more time due to the integration of the extra steps, which contributes to refined skeletons. All the tests are carried out on a PC with an Intel Core i7-4790K CPU and 16GB RAM.

In Fig. 7, we extract skeletons from models with thin cylindrical components. Note that these models exhibit significantly heavy noise. In these examples, both our method and algorithm [15] obtain satisfactory results, however, our method requires less tedious parameter tuning to produce the desirable results. That is, our method needs less trial and error operations to get the final results. Without distance field guidance, using the algorithm [15], the projected samples are apt to deviate from the center of components without careful parameters tuning, especially for the support radius h .

Impact of sampling density As having been pointed out, before voxelization, we need to detect the sparsely sampled regions and upsample these regions on the input point cloud Q . We define local sampling density as $\rho_i = k/(r_i)^3$. If ρ_i is below a given threshold value, then that region is of insufficient sampling density and need upsampling. In our implementation, we set $k = 5$ and set r_i with a default neigh-

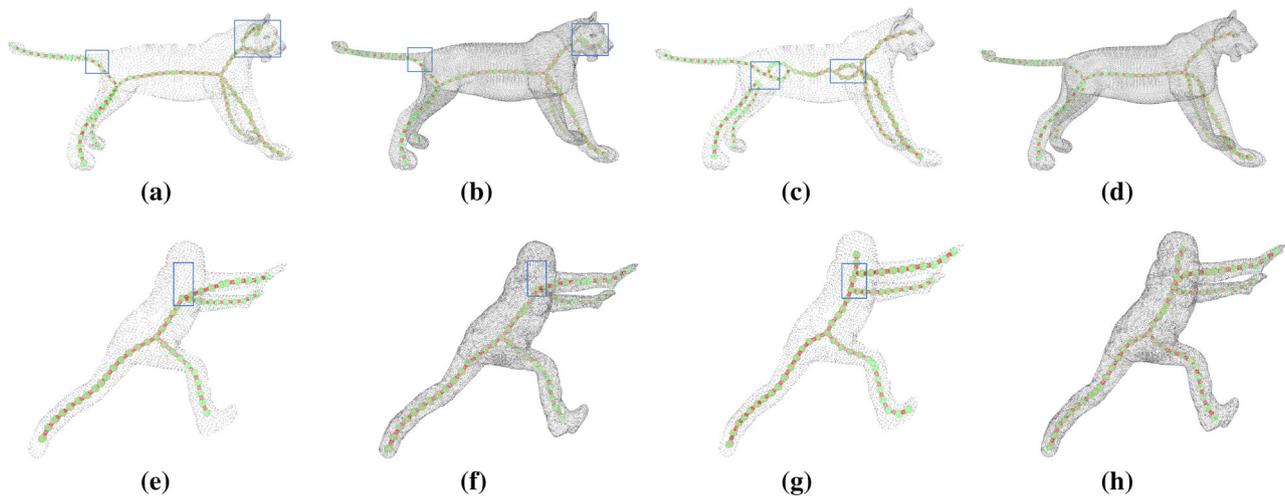


Fig. 8 Skeletons extracted from models before and after upsampling. **a, e** Skeletons from original models by [15]. **b, f** Skeletons from upsampled models by [15]. **c, g** Skeletons from original models by our method. **d, h** Skeletons from upsampled models by our method

neighborhood size $r_i = 3d_{bb}/\sqrt[3]{|J|}$, where d_{bb} is the diagonal length of Q 's bounding box and $|J|$ is the number of points in Q . So the required sampling density ρ_i is $5|J|/27d_{bb}^3$. For above neighborhood size, if the neighbors contained in the neighborhood are less than 3, then we increase neighborhood size to make the interpolation-based upsampling work, and produce enough samples for voxelization. Actually, we usually make the sample points denser than above threshold to facilitate the subsequent voxelization.

We perform our algorithm on two models with sparsely sampled torso and their upsampled models (see Fig. 8). When extracting skeletons from the sparse models containing 6381/4211 points, the results of both [15] and our method are not satisfactory. For our method, as insufficient sampling density will lead to the disconnection of distance field with high voxelization resolution, which leads to the artifacts in the final skeleton. After upsampling the models (amounting to 33,920/22,802 points), our method obtains compelling results, which are much better than the results of [15].

Comparison to mesh contraction We compare our method with Oscar Kin-Chung Au's method on the dog model provided by the authors [2], and show respective results as follows. Note that we perform Oscar's method on the mesh model using the executive files presented by the authors. We perform our method on the corresponding point cloud. As illustrated in following Fig. 9, the result of Oscar has more skeleton branches around the head part of the dog, and the skeleton retains the key features of the head part. In comparison, our method only retains the most salient features of the model. We also compare our result with that of [15].

Comparison to L_1 -medial skeleton In Fig. 10, we work on the models with different characters, and also compare with [15]. In these examples, each model has some components

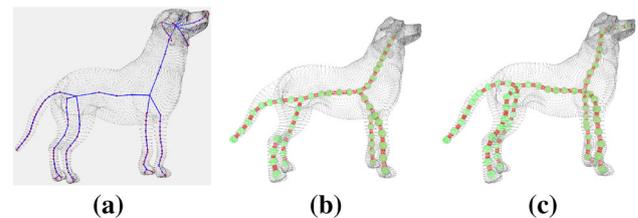


Fig. 9 Result comparisons. **a** Result of Oscar [2], **b** result of [15], **c** our result

that are not thin and cylindrical. In performing iterative contraction, Huang et al. [15] have to gradually increase the support radius h to further contract non-branch points to produce skeletons. While setting h in this way is model-special, to process complex model, the sample points may not contract to the component center. In our experiments, we found that without prior for the centers of components, the L_1 median projection, aimed to automatically contract the points to the centers, sometimes does not work well. Furthermore, to receive satisfactory results, besides the radius h setting, appropriate bridge points selection is also a non-trivial problem. Without model shape prior, wrong bridge points may be selected, which will make the resulting skeleton penetrate out the model. However, with the guide of the distance field, our method provides much better results and also requires less trial and error operations. In Fig. 10, we present comparison results on six models. Note that we use the code presented by authors of [15] to produce all the results of the method [15], and for each result, we use the optimal parameters and try to obtain the best results.

Compared with [15], our method also has some disadvantages. Even for models with significant heavy outlier, the method [15] can be performed on these models without

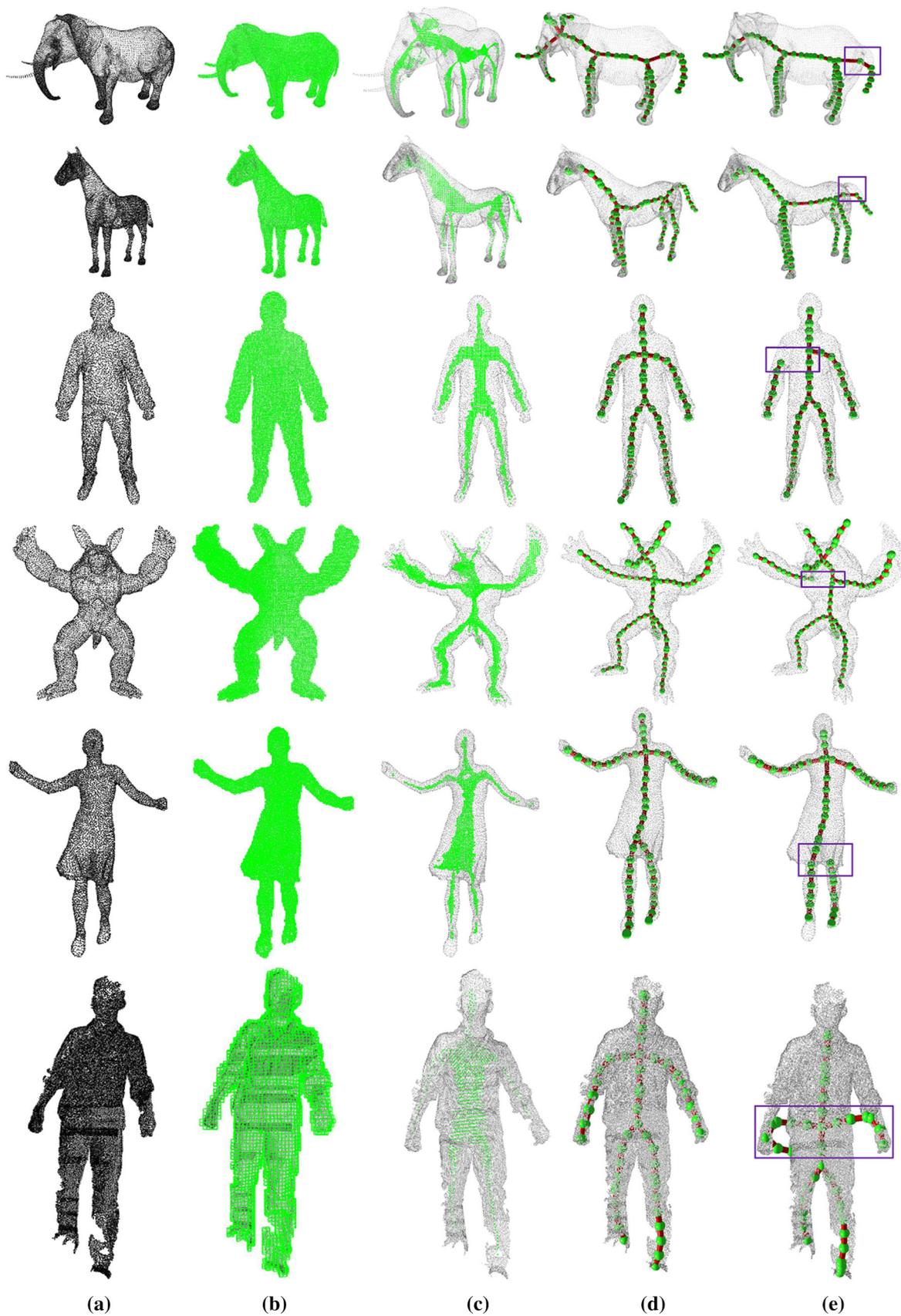


Fig. 10 Skeleton extraction comparisons. **a** Input models, **b** voxelization results, **c** initial skeletons, **d** our final skeletons, **e** results of [15]

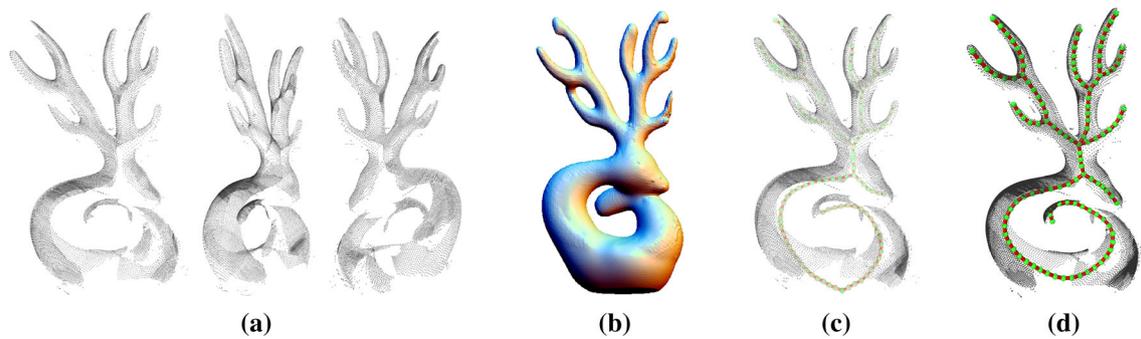


Fig. 11 Failed example. **a** Input modal viewed from 3 viewing directions: front, *left side* and *right side*. **b** RBF reconstruction result of **(a)**. **c** Skeleton extracted from **(b)** using our method. **d** Skeleton extraction result of [15] (result from the authors' paper)

preprocessing for outlier removal. In this situation, we need to pre-filter the models using the locally optimal projection (LOP) [23] before performing effective voxelization. While it also should be pointed out that, the preprocess procedure will not alter the topology of the input model, thus, it will not have effect on the final skeleton generation.

Limitations Although our method can obtain excellent results for models with complex shape, our method still has the following limitations. The proposed scheme works well for most input data, however, for those scanned point clouds whose missing regions are too large, our method would be challenged. Taking the deer model in Figure 1 of [15] as an example, our method performs worse or even does not work. As illustrated in the Fig. 11a, large missing regions are distributed on the model. Our method cannot perform voxelization on the input model, thus, fails to extract the skeleton from the input model. If resample and complete the model using RBF method (Fig. 11b), we can voxelize the model and extract the skeleton, however, as illustrated in Fig. 11c, the skeleton deviates from the original model part center, since it is extracted from the upsampled models obtained using energy minimization model (RBF). The method of [15] can extract satisfactory result from the original input model without upsampling (Fig. 11d). This example illustrates that our method usually fails to extract the satisfactory skeleton from the model with large missing surface regions.

Finally, compared with [15], our method has higher time complexity for computing the distance field, and needs more memory consumption for voxelization presentation.

8 Conclusion

In this paper, we present a distance field guided L_1 -median to extract skeletons from the point cloud. Our method combines the advantages of both the distance field based method and of L_1 -median based method, and particularly, improves

the robustness and effectiveness of the L_1 -median skeleton method. We have extracted the skeletons from a variety of point clouds to illustrate the effectiveness of the proposed methods.

Many 1D skeleton extraction algorithms have been proposed, and every algorithm has its advantages and disadvantages. As skeleton extraction from complex point cloud data, especially from the scanned raw data with missing data, is an ill-posed problem, our method also exhibits some limitations. Thus, we believe that our method can be considered as a good complement for the skeleton extraction community.

Compared with extracting skeleton from one static model, extracting the skeleton series from the time-varying surface sequences is a more challenging task. As desirable extracted skeleton series should not only be topologically clean but also be temporally coherent. In the future, we would like to extend our method to this research direction.

Acknowledgements This work was partly supported by NSFC (No.614 72288, No. 61672390, No. 41201404, No. U1536204), NCET (NCET-13-0441) and the Key Grant Project of State Key Lab of Software Engineering (SKLSE-2015-A-05).

References

1. Alexa, M., Behr, J., Cohen-Or, D., Fleishman, S., Levin, D., Silva, C.T.: Computing and rendering point set surfaces. *IEEE Trans. Vis. Comput. Graph.* **9**(1), 3–15 (2003)
2. Au, O.K.C., Tai, C.L., Chu, H.K., Cohen-Or, D., Lee, T.Y.: Skeleton extraction by mesh contraction. *ACM Trans. Graph. (TOG)* **27**(3), 44 (2008)
3. Avron, H., Sharf, A., Greif, C., Cohen-Or, D.: ℓ_1 -sparse reconstruction of sharp point set surfaces. *ACM Trans. Graph. (TOG)* **29**(5), 135 (2010)
4. Bitter, I., Kaufman, A.E., Sato, M.: Penalized-distance volumetric skeleton algorithm. *IEEE Trans. Vis. Comput. Graph.* **7**(3), 195–206 (2001)
5. Bouix, S., Siddiqi, K.: Divergence-based medial surfaces. In: *Computer Vision-ECCV*, Springer, pp. 603–618 (2000)
6. Bucksch, A., Lindenberg, R., Menenti, M.: Skeltre. *Vis. Comput.* **26**(10), 1283–1300 (2010)

7. Cao, J., Tagliasacchi, A., Olson, M., Zhang, H., Su, Z.: Point cloud skeletons via laplacian based contraction. In: Shape Modeling International Conference (SMI), pp. 187–197. IEEE (2010)
8. Chuang, J.H., Ahuja, N., Lin, C.C., Tsai, C.H., Chen, C.H.: A potential-based generalized cylinder representation. *Comput. Graph.* **28**(6), 907–918 (2004)
9. Chuang, M., Kazhdan, M.: Fast mean-curvature flow via finite-elements tracking. In: *Computer Graphics Forum*, vol. 30, pp. 1750–1760. Wiley Online Library (2011)
10. Cornea, N.D., Silver, D., Min, P.: Curve-skeleton properties, applications, and algorithms. *IEEE Trans. Vis. Comput. Graph.* **13**(3), 530–548 (2007)
11. Dey, T.K., Sun, J.: Defining and computing curve-skeletons with medial geodesic function. In: *Symposium on Geometry Processing*, pp. 143–152 (2006)
12. Gagvani, N., Silver, D.: Parameter-controlled volume thinning. *Graph. Models Image Process.* **61**(3), 149–164 (1999)
13. Hassouna, M.S., Farag, A.A.: Robust centerline extraction framework using level sets. In: *Computer Vision and Pattern Recognition, CVPR. IEEE Computer Society Conference on*, vol. 1, pp. 458–465. IEEE (2005)
14. Huang, H., Li, D., Zhang, H., Ascher, U., Cohen-Or, D.: Consolidation of unorganized point clouds for surface reconstruction. *ACM Trans. Graph. (TOG)* **28**(5), 176 (2009)
15. Huang, H., Wu, S., Cohen-Or, D., Gong, M., Zhang, H., Li, G., Chen, B.: L1-medial skeleton of point cloud. *ACM Trans. Graph.* **32**(4), 65 (2013)
16. Jalba, A.C., Sobiecki, A., Telea, A.C.: An unified multiscale framework for planar, surface, and curve skeletonization. *IEEE Trans. Pattern Anal. Mach. Intell.* **38**(1), 30–45 (2016)
17. Jiang, W., Xu, K., Cheng, Z.Q., Martin, R.R., Dang, G.: Curve skeleton extraction by coupled graph contraction and surface clustering. *Graph. Models* **75**(3), 137–148 (2013)
18. Katz, S., Tal, A.: Hierarchical mesh decomposition using fuzzy clustering and cuts, vol. 22. ACM (2003)
19. Kustra, J., Jalba, A., Telea, A.: Computing refined skeletal features from medial point clouds. *Pattern Recognit. Lett.* **76**, 13–21 (2016)
20. Li, G., Liu, L., Zheng, H., Mitra, N.J.: Analysis, reconstruction and manipulation using arterial snakes. *ACM Trans. Graph. TOG* **29**(6), 152 (2010)
21. Li, X., Woon, T.W., Tan, T.S., Huang, Z.: Decomposing polygon meshes for interactive applications. In: *Proceedings of the 2001 symposium on Interactive 3D graphics*, pp. 35–42. ACM (2001)
22. Liao, B., Xiao, C., Jin, L., Fu, H.: Efficient feature-preserving local projection operator for geometry reconstruction. *Comput. Aided Des.* **45**(5), 861–874 (2013)
23. Lipman, Y., Cohen-Or, D., Levin, D., Tal-Ezer, H.: Parameterization-free projection for geometry reconstruction. *ACM Trans. Graph. (TOG)* **26**(3), 22 (2007)
24. Livny, Y., Yan, F., Olson, M., Chen, B., Zhang, H., El-Sana, J.: Automatic reconstruction of tree skeletal structures from point clouds. *ACM Trans. Graph. (TOG)* **29**(6), 151 (2010)
25. Malandain, G., Fernández-Vidal, S.: Euclidean skeletons. *Image and vision computing* **16**(5), 317–327 (1998)
26. Natali, M., Biasotti, S., Patanè, G., Falcidieno, B.: Graph-based representations of point clouds. *Graph. Models* **73**(5), 151–164 (2011)
27. Pang, Z., Zhao, Y., Xiao, C.: Effective skeletons extraction for animated surfaces based on geometry propagation. *Comput. Anim. Virtual Worlds* **26**(3–4), 301–309 (2015)
28. Preiner, R., Mattausch, O., Arikian, M., Pajarola, R., Wimmer, M.: Continuous projection for fast I1 reconstruction. *ACM Transactions on Graphics (Proc. of ACM SIGGRAPH)* **33**(4), 47:1–47:13 (2014)
29. Sethian, J.A.: Fast marching methods. *SIAM Rev.* **41**(2), 199–235 (1999)
30. Sharf, A., Lewiner, T., Shamir, A., Kobbelt, L.: On-the-fly curve-skeleton computation for 3d shapes. *Comput. Graph. Forum* **26**(3), 323–328 (2007)
31. Siddiqi, K., Pizer, S.: *Medial representations: mathematics, algorithms and applications*, 1st edn. Springer Publishing Company Incorporated, New York (2008)
32. Small, C.G.: A survey of multidimensional medians. *International Statistical Review/Revue Internationale de Statistique* pp. 263–277 (1990)
33. Tagliasacchi, A., Alhashim, I., Olson, M., Zhang, H.: Mean curvature skeletons. *Comput. Graph. Forum* **31**(5), 1735–1744 (2012)
34. Tagliasacchi, A., Zhang, H., Cohen-Or, D.: Curve skeleton extraction from incomplete point cloud. *ACM Trans. Graph. (TOG)* **28**(3), 71 (2009)
35. Verroust, A., Lazarus, F.: Extracting skeletal curves from 3d scattered data. In: *Shape Modeling and Applications, 1999. Proceedings. Shape Modeling International '99*, pp. 194–201 (1999). doi:[10.1109/SMA.1999.749340](https://doi.org/10.1109/SMA.1999.749340)
36. Zheng, Q., Hao, Z., Huang, H., Xu, K., Zhang, H., Cohen-Or, D., Chen, B.: Skeleton-intrinsic symmetrization of shapes. In: *Computer Graphics Forum, Wiley Online Library*, vol. 34, pp. 275–286 (2015)
37. Zhou, Y., Toga, A.: Efficient skeletonization of volumetric objects. *IEEE Trans. Vis. Comput. Graph.* **5**(3), 196–209 (1999). doi:[10.1109/2945.795212](https://doi.org/10.1109/2945.795212)



Chengfang Song received the Ph.D. degree from the State Key Laboratory of CAD & CG, Zhejiang University in 2007. He is currently an assistant professor of Wuhan University, China. His research interests include 3D reconstruction and visualization.



Zhiqiang Pang is a master student at the School of Computer, Wuhan University. His research interests include digital geometry processing.



Xiaoyuan Jing received the Doctoral degree of Pattern Recognition and Artificial Intelligence from the Nanjing University of Science and Technology, 1998. He was a Professor with the Department of Computer, Shenzhen Research Student School, Harbin Institute of Technology, 2005, and a Research Fellow with the Department of Computing, Hong Kong Polytechnic University. Now he is a Professor with the State Key Laboratory of Software Engineering, School of

Computer, Wuhan University, China. His research interests include pattern recognition, machine learning, software engineering and artificial intelligence.



Chunxia Xiao received his BSc and MSc degrees from the Mathematics Department of Hunan Normal University in 1999 and 2002, respectively, and received his Ph.D. from the State Key Lab of CAD & CG of Zhejiang University in 2006, China. He is currently a professor at the Computer School, Wuhan University, China. His research interests include digital geometry processing, image and video processing, and computational photography.