Point Cloud Skeletons via Laplacian-Based Contraction

Junjie Cao*, Andrea Tagliasacchi†, Matt Olson†, Hao Zhang† and Zhixun Su*
*Dalian University of Technology, Email: jjcao1231@gmail.com, zxsu@comgi.com
†Simon Fraser University, Email: {ata2,haoz,matto}@cs.sfu.ca

***Abstract*—We present an algorithm for curve skeleton extraction via Laplacian-based contraction. Our algorithm can be applied to surfaces with boundaries, polygon soups, and point clouds. We develop a contraction operation that is designed to work on generalized discrete geometry data, particularly point clouds, via local Delaunay triangulation and topological thinning. Our approach is robust to noise and can handle moderate amounts of missing data, allowing skeleton-based manipulation of point clouds without explicit surface reconstruction. By avoiding explicit reconstruction, we are able to perform skeleton-driven topology repair of acquired point clouds in the presence of large amounts of missing data. In such cases, automatic surface reconstruction schemes tend to produce incorrect surface topology. We show that the curve skeletons we extract provide an intuitive and easy-to-manipulate structure for effective topology modification, leading to more faithful surface reconstruction.**

***Keywords*-curve skeleton; point cloud; Laplacian; contraction; topology repair; surface reconstruction**

I. Introduction

The curve skeleton of a 3D model is a 1D representation that intuitively captures the model's topological characteristics. Applications which utilize curve skeletons for shape analysis and processing include segmentation ([1], [2], [3], [4]), object matching and retrieval ([5], [6], [7]), surface reconstruction ([8], [9]), and animation ([4], [10], [11], [12], [13], [14]), among others. Curve skeletons are not the only means for obtaining a topological representation of a shape; another notable example is the medial axis [15], which provides a powerful dual representation of the shape surface. However, the medial axis of a nontrivial 3D shape is typically composed of a set of interconnected sheets which are difficult to model and manipulate.

Existing works on curve skeleton extraction almost exclusively operate on complete surface models ([3], [4], [12], [16], [17]) and they can produce excellent results. However, geometry representations we frequently encounter may not always be watertight. In particular, most interesting models are first acquired in the form of point cloud data. As well, models with boundaries and even triangle soups can often be found in shape databases [18]. While one can define the curve skeleton of a watertight mesh as a subset of the medial axis [3], the medial axis of a point set is not well defined. This poses a significant obstacle to implementing applications such as shape matching, retrieval, segmentation, or animation on point cloud data without a costly reconstruction phase.

Indeed, one may compute the curve skeleton of a point set by first performing a surface reconstruction and then apply mesh skeletonization. State-of-the-art surface reconstruction techniques ([19], [20], [21], [22]) can provide high quality results when the underlying surface is sufficiently well-sampled. However, self-occlusion and other acquisition hindrances often generate under-sampled regions; this is an especially persistent issue during dynamic or time-varying shape acquisition ([23], [24], [25], [26]). Under these circumstances, results from even the best surface reconstruction algorithm can be error-prone, leading to incorrect surface topology which in turn cause the extracted skeletons to be erroneous, as in Fig. 1.

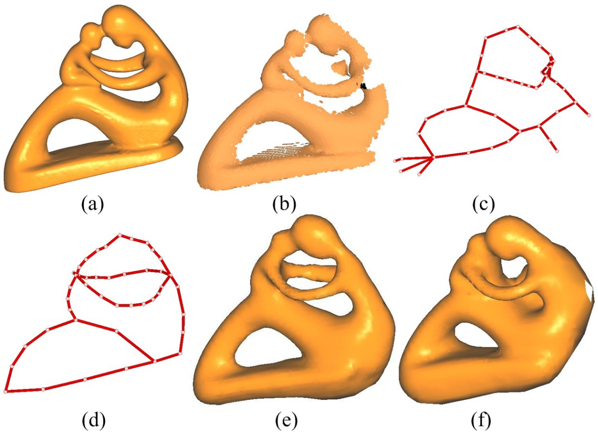


Figure 1. Point cloud skeleton and skeleton-assisted topology repair and surface reconstruction. (a) Original model. (b) Input point cloud with a great deal of missing data. (c) Curve skeleton extracted via Laplacian-based contraction, while descriptive, contains topological errors. After simple user operations to repair the skeleton (d), topologically correct surface reconstruction is obtained (e), compared to the result of Poisson reconstruction (f) from the input point cloud (b).

In these cases, the user may be able to manually correct the topology imposed by the interpolation [27]. However, reconstruction algorithms often produce high-resolution surface models — editing an improperly reconstructed surface thus requires the user to interact with a large number of polygons. An interesting thought here would be to rely on curve skeletons, which provide simple and intuitive characterizations of shape topology and are much easier to interact with, to assist in surface reconstruction. Indeed, fixing the topology of a shape by editing its skeleton seems to be a more viable option ([28], [29]). This, however, requires one to extract a curve skeleton directly from a point cloud.

In this paper, we propose an algorithm to construct the curve skeleton directly on point cloud data, without computing the surface of the object. In order to achieve this, we perform a contraction procedure inspired by the recent

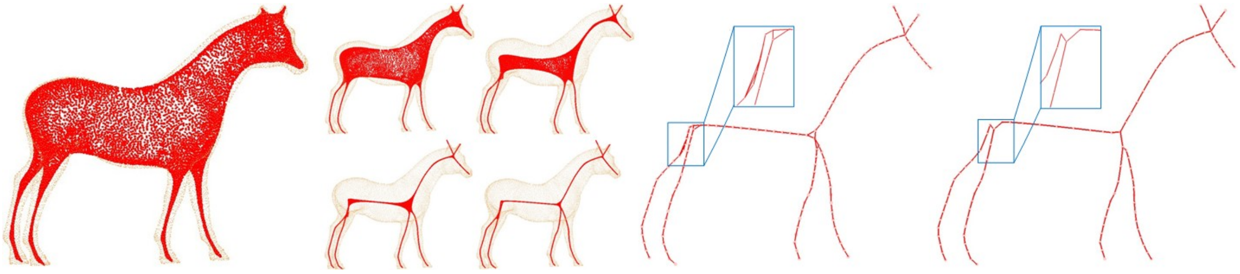


Figure 2. Overview of our algorithm. (a) Input point cloud (orange) and the contracted points (red) after 1 iteration. (b) Contracted points after 2-5 iterations. (c) Skeleton graph constructed by farthest-point sampling (light red points) with a fixed radius ball and connectivity (red lines) inherited from the local neighborhood relationship of the input points. (d) The final 1D curve skeleton after topology thinning.

work of Au *et al.* [4], but on point cloud data; see Fig. 2(a) and (b). This is followed by topological thinning to reduce the skeleton structure to a line; see Fig. 2(c) and (d). We show that our contraction is robust to noise and can deal with moderate amounts of missing data. However, as the contraction relies on a local analysis, the extracted skeleton can be imprecise or even incorrect in the presence of large amount of missing data. In this scenario, the scanned data carries insufficient geometric information to determine the correct topology without first imposing a prior on the shape. Fortunately, the extracted skeleton provides a compact editing structure which can be employed to correct the model's topology. We thus develop a skeleton-assisted topology-preserving reconstruction procedure based on the finite element method of Sharf *et al.* [27].

Note that while our method is broadly similar to that of Au *et al.* [4], we extend their work in several important ways. Foremost among these is our ability to handle point cloud models, meshes with boundaries, triangle soup inputs without global connectivity, and even missing data from incomplete range scans. We also improve upon the determination of parameters and iteration termination conditions in the contraction process. Beside adapting the method to point cloud models, we lower the number of iterations necessary to produce an intermediate point set suitable for topological thinning. In addition, their connectivity-surgery method is not suitable for our case. Instead, we develop a robust topological thinning method to convert the contracted point cloud into a curve skeleton.

In summary, we present a robust curve skeleton extraction algorithm for point clouds and apply it to an interactive topology-driven reconstruction scheme. We demonstrate experimentally that the extracted skeletons have high quality despite having moderate amounts of missing data in the input. In particular, we show that our method produces better results on point scans with missing data than a state-of-the-art mesh skeletonization algorithm applied to a reconstructed mesh.

II. Related work

Skeleton extraction has been the focus of extensive research dating back to the the 1960s. While existing algorithms target a variety of model characteristics and applications, extraction procedures can be roughly divided into volumetric methods and geometric methods [30].

Volumetric methods require a voxelization of the input model [31]. Some iteratively remove a model's simple points [32], thinning the structure until a curve skeleton is obtained [33], while others trace and prune ridges of the distance field. Spatial discretization may lead to loss of detail and numerical instability at low resolutions, and in general it requires the inside of the shape to be well defined. This is clearly difficult for point clouds. An exception is the algorithm of Sharf *et al.* [34], where, following the Euclidean distance field, a dynamic mesh is expanded inside the point cloud tracing the shape skeleton. In this solution, complete water-tightness is not required and extraction is successful even with moderate amount of missing data. However, this requires a careful setting of the tension parameter to prevent the dynamic mesh from leaking outside the shape.

Rather than voxelize the input, *geometric methods* work directly on triangle meshes or point clouds. A noteworthy solution is to employ Voronoi diagrams ([3], [35]) and extract a curve skeleton by pruning the extracted approximation of the medial axis. However, such constructs are sensitive to sampling conditions and noise which are often not met in real scanned data. On the other hand, if only a topological representation of the shape is desired, Reeb graph-based methods [36] provide a viable solution that produces curve skeleton directly. Given a real-valued function defined on the model's surface, Reeb graphs cluster approximate iso-value regions and connect them to each other constructing a graph which can be embedded in the 3D space. The choice of this function distinguishes the various methods. The height function in [37] is sensitive to the model's orientation, while both the geodesic function [36] and the harmonic function [38] require the user to specify proper boundary conditions. Verroust and Lazarus [39] generalize the Reeb graph-based method to unorganized points, but the skeleton extracted does not contain all topological features and requires further processing. Recently a novel geometric method [9] is proposed, based on a generalized rotationally symmetric analysis. This is often suitable for curve skeleton extraction but is not generalizable to all shapes.

The technique which is most related to ours is the mesh contraction solution of Au *et al.* [4], where the volume-reducing characteristics of Laplacian smoothing ([40], [41]) are exploited to perform robust curve skeleton extraction for closed triangular meshes. However, to apply this method to a point set we would need to apply surface reconstruction. We consider this to be computationally unnecessary, as reconstruction aims to recover details while curve skeleton extraction focuses on global properties of the shape. Furthermore, as shown in Fig. 6(e), incorrect reconstruction may lead to incorrect skeleton extraction.

Classical reconstruction ([21], [42], [43]) can solve, by interpolation, many of the problems caused by smaller amounts of missing data. However, even mostly well-sampled surfaces may contain critical regions which are incorrectly reconstructed, typically resulting in topological tunnels. Techniques for topology correction have been developed to compensate, either based on skeletons [28], local scribbles [27] or sketching [29]. While both [28] and [29] require a watertight mesh as input, Sharf *et al.* [27] integrates topology correction directly into the reconstruction procedure. Our skeleton-driven reconstruction method can be seen as a combination of the three solutions above: it uses a curve skeleton to specify topology, which provides scribble-like information to guide the reconstruction technique. In addition, our skeletons present some similarity to the sketches used in [29]; however, they are computed automatically, without requiring the user to specify the full shape topology from scratch.

III. Curve skeleton extraction via Laplacian-based contraction

Our algorithm proceeds as follows: Given a point cloud $P = \{p_i\}$, we first contract P to a zero-volume point set C as described in Section III-A. Next, we build a skeleton graph G using sub-samples of C. We then contract edges of G iteratively to build a skeleton T, which is a curve skeleton with cycles. This process is illustrated in Section III-B. Finally, to restrict T to the interior of P as much as possible, we move every vertex of T to the center of its local neighborhood in P as described in [4].

A. Geometric Contraction

The geometric contraction operation first presented in [4] produces excellent results on triangle meshes. We extend the algorithm to support point cloud models. The contraction maintains the global shape of the input model by anchoring points chosen by an implicit Laplacian smoothing process. Thus, the contracted point cloud C captures the geometric characteristics of the input with minimal volume, offering an initial estimation of the position of the final curve skeleton. Here we briefly describe the process and indicate our adaptations.

As stated in [4], the contraction is computed by iteratively solving the linear system:

$$\begin{bmatrix} W_L L \\ W_H \end{bmatrix} P' = \begin{bmatrix} 0 \\ W_H P \end{bmatrix}, \tag{1}$$

where L is a $n \times n$ Laplacian matrix with cotangent weights, P' is a contracted point cloud, and W_L and W_H are the diagonal weight matrices balancing the contraction and attraction constraints, the ith diagonal element of W_L (resp. W_H) is denoted $W_{L,i}$ (resp. $W_{H,i}$). The solution to this problem minimizes the quadratic energy [44]:

$$\|W_L L P'\|^2 + \sum_i W_{H,i}^2 \|p_i' - p_i\|^2, \tag{2}$$

where the first term removes geometry details along the normal directions using implicit Laplacian smoothing and the second preserves shape geometry during contractions.

Laplacian operator construction for point cloud data: The recent introduction of the Voronoi-Laplacian [45] and PCD-Laplacian [46] addressed the definition of a suitable operator for point cloud data. These two solutions are similar, as both need to compute a planar Delaunay triangulation of the set of points within distance R and then use the triangulation to define a weighting scheme. Since no automatic scheme for the choice of the scale parameter R is available, applying such techniques directly would be difficult, especially as we need to re-compute the Laplacian at every iteration. In addition, large values of R might affect the sparseness of the matrix L and slow down the contraction process.

To mitigate these problems, we first extract an approximate neighborhood of P_i by computing its k nearest neighbors $N_k(P_i)$ and projecting them on the tangent plane defined by their principal components. We then construct a planar Delaunay triangulation and define the one-ring neighbors of P_i as those participating in the definition of the Laplace operator. As the contraction process progresses, the computation of a correct tangent plane becomes difficult; to tackle this problem, we compute neighborhood information only at the first iteration. Consequently, subsequent iterations will only need to update the Laplacian weights, rather than the whole operator.

Many alternatives are available for the choice of the weighting scheme, such as combinatorial weights [44], mean value weights [47] or cotangent weights [48]. Fig. 3 clearly illustrates how even for point clouds, the cotangent weighting scheme achieves superior contraction quality, similarly to the mesh scenario, where zeroing the Laplacian would cause a contraction in the direction of the mean curvature normal [48].

Point cloud Laplacian parameters: The only parameter required for our discrete point cloud Laplacian is the number of nearest neighbours k, used to construct the domain over which we estimate the tangential plane. Throughout our experiments, we set $k = .012\#_{samples}$, restricting the values to the range $[8..30]$. This narrow range is justified by the fact that we only need a 1-ring set of samples for the construction of the Laplacian.

Iteration and contraction parameters: To collapse P into a curve skeleton, we solve equation (1) iteratively while updating W_L and W_H. P' is contracted noticeably even after the first iteration. To contract P' further, we

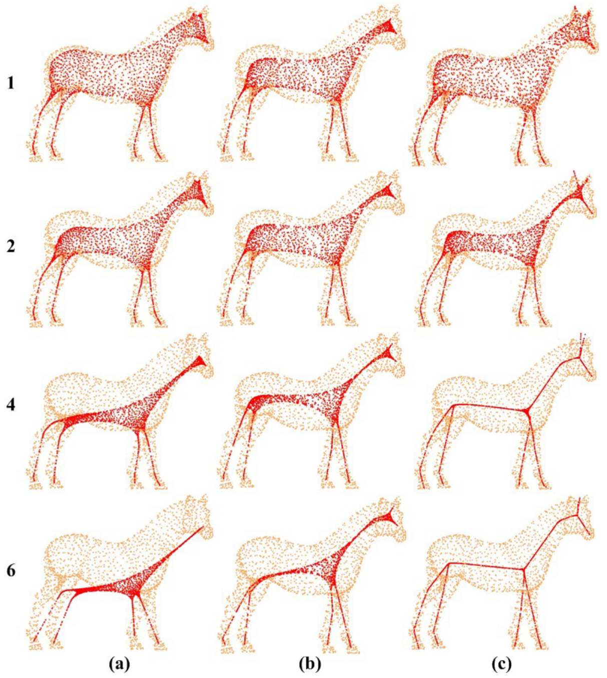


Figure 3. Comparison of the contraction techniques built using various types of point cloud Laplacian. Using cotangent weights requires fewer iterations and better preserves the geometry of the input shape. (a) Laplacian with combinatorial weights. (b) Laplacian with tangent weights. (c) Our solution using tangent Delaunay Laplacian with cotangent weights. The first column is the iteration number.

increase the contraction weights W_L after each iteration. To avoid over-contraction, we must also update the attraction weights $W_{H,i}$ according to p_i's collapsed degree. This is determined by the extent of of its neighbors, which we approximate by $\min_{q \in N_k(p_i)} \|p_i - q\|$. Specifically, we want points with smaller neighborhoods to be attracted more strongly to their current positions.

We evaluate iteration t as follows:

1) Solve $\begin{bmatrix} W_L^t L^t \\ W_H^t \end{bmatrix} P^{t+1} = \begin{bmatrix} 0 \\ W_H^t P^t \end{bmatrix}$ for P^{t+1},

2) Update $W_L^{t+1} = s_L W_L^t$ and $W_{H,i}^{t+1} = W_{H,i}^0 S_i^0 / S_i^t$, where S_i^t and S_i^0 are the current and original neighborhood extent of point i, respectively,

3) Compute the new Laplacian operator L^{t+1} with the current point cloud P^{t+1}.

This process stops when the average value of $(S_i^{t+1} - S_i^t)/S_i^0 < t$, where t is a user-specified threshold. (We set $t = 0.01$ for all models in this paper.) This indicates convergence to a steady state.

We find that setting $s_L = 3.0$ allows us to converge in fewer than 5 iterations on most models. Since the scale of the Laplacian coordinate is proportional to a point's neighborhood extent (under the same local neighborhood shape), the contraction forces from the Laplacian equations are smaller for denser models. Thus to handle models of different sizes and resolutions, we set $W_L^0 = 1/(5S^0)$ and $W_H^0 = 1.0$ for all results in this paper, where S^0 is the mean neighborhood extent of the point cloud.

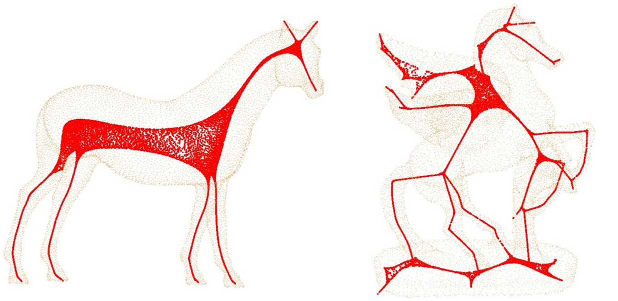


Figure 5. Methods to convert a point cloud to a curve skeleton should be robust against the two challenging cases constructed using less iterations or poor contraction parameters. (Left) The horse model after three contractions. (Right) The final contracted Pegaso model with the initial attraction weight $W_H^0 = 10$; the skeleton is shown in Fig. 4(h).

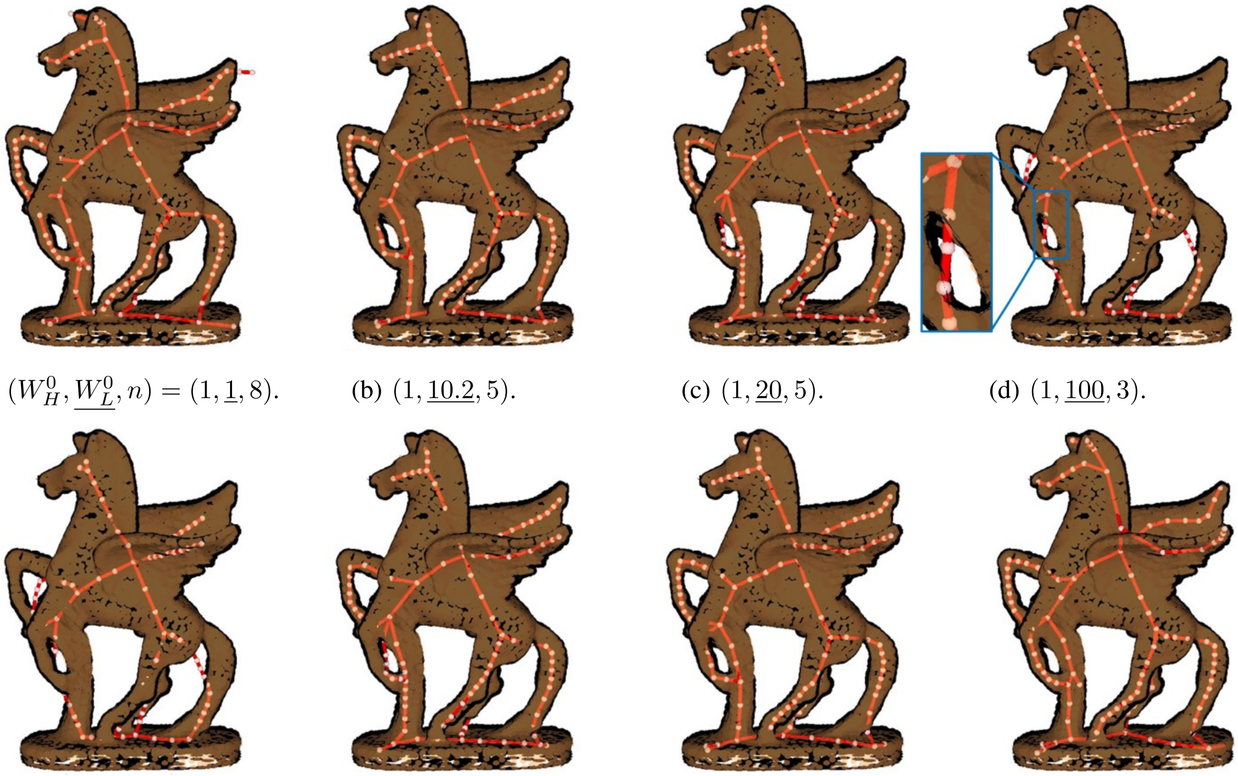
Parameters: The key to contraction is breaking the balance of the forces in W_H and W_L. The initial ratio between the two is important. If the initial contraction weights are too small, skeleton nodes near the tips of the model are prone to extend out of the model, as near the tip of the Pegaso's wing in Fig. 4(a). If the contraction weights are too large, the strong contraction force of the thicker region may attract the contracted points of the nearby thinner part towards the thicker part, forcing those points out of the model as with the tail of the Pegaso in Fig. 4(d). High contraction forces can also collapse branches around a small tunnel, such as that near the left front leg of the Pegaso in Fig. 4(d). Note also that the iteration times decrease with the initial contraction weight. The initial attraction weight has an opposite effect to the initial contraction weight, shown in Fig. 4(e-h). Increasing W_L^0 or decreasing W_H^0 both result in nearly the same skeleton with only small differences as shown in Fig. 4(a) and (h). This stems from the fact that the updated weights in each iteration will change the initial ratio. Fig. 4 also shows that small changes to the initial contraction and attraction weights slightly affect the resulting structure.

Stability of the iterative contraction: The iterative increase of the attraction weights W_H makes the system matrix more diagonally dominant as the iteration progresses; this ensures stability. During the process, we also avoid any possible numerical errors, including infinity values and divide-by-zero error as in [4]. The result of this algorithm is a thin shape C which approximates the curve skeleton; see Fig. 2(c).

B. Topological thinning

The result C of the previous step is a point cloud, not a 1D curve skeleton. Previous methods to convert a point cloud to a curve skeleton, such as [49] and [9], require joint identification and delicate parameter tuning to handle a contracted shape with different local shape characteristics, as in Fig. 5(b). They are also not as robust as the method we present here, particularly when the input shape C contains thick regions such as Fig. 5(a). Our basic idea is similar to methods first presented in [1].

Building connectivity: We build a skeletal structure from C first by imposing an initial connectivity, then by



(a) $(W_H^0, \underline{W_L^0}, n) = (1, \underline{1}, 8)$. (b) $(1, \underline{10.2}, 5)$. (c) $(1, \underline{20}, 5)$. (d) $(1, \underline{100}, 3)$.

(e) $(\underline{W_H^0}, W_L^0, n) = (\underline{0.1}, 10.2, 3)$. (f) $(\underline{0.5}, 10.2, 5)$. (g) $(\underline{1}, 10.2, 5)$. (h) $(\underline{10}, 10.2, 8)$.

Figure 4. Our algorithm is robust to small changes in the default ratio of contraction and attraction forces. In the first row we fix W_H^0 to the default value of 1 and vary W_L^0. In the second row, we fix W_L^0 to the default value of 10.2 and vary W_H^0. For each image we list W_H^0, W_L^0 and the iteration count n required to produce the curve skeleton.

applying an edge-contraction operation. We first sample C using farthest-point sampling and a ball of radius r. Each sample $g_i \in G$ represents the set of associated points C_i in C which are closest to it, such that $C = \bigcup_i C_i$ and $C_i \bigcap C_j = \emptyset, \forall i \neq j$. Connecting g_i and g_j if their associated points share common local 1-ring neighbors, we obtain a graph G with uniformly distributed nodes as shown in Fig. 2(c).

Edge contraction: Given this connectivity, we collapse unnecessary edges until no triangles exist to build a curve skeleton. To distribute the final skeleton vertices uniformly and capture the shape of G, at each step we collapse the edge with minimum Euclidean length to its midpoint, and triangles incident to the edge are removed. The points associated with the two endpoints are assigned to the newly created vertex. We iterate this procedure until all triangles have been removed and a 1D curve skeleton T computed as shown in Fig. 2(d).

Limitations: Our topological contraction algorithm is able to retain the topology except in the following two cases. As we are considering unorganized point clouds, we cannot rely on connectivity to distinguish nearby structures. If the distance between two neighboring structures is smaller than the radius r, then points which belong to two different branches may be collapsed into one. Alternatively, if r is larger than the diameter of a hole, then the hole may be collapsed to a single point while computing the initial graph for topological thinning. For our results, we consistently choose the value of r as 0.02 times the length of the longest diagonal of the model's bounding box, but an adaptive solution which considers a sampling proportional to the local feature size [50] might achieve results across multiple scales of geometric detail.

IV. Results

We show some of our skeleton extraction results in Fig.s 1, 2, and 4. Here we present a broader cross-section of results, showing the performance of our curve skeleton extraction from surfaces with boundaries, polygon soups and complete and incomplete point clouds. For visualization, we render the back-facing points in black and front facing points using transparent colored splats except for Fig. 2, 5, and 9, which are rendered in MATLAB directly. The final skeleton is shown in red.

A. Curve skeleton from incomplete point clouds

If the point cloud is complete, applying Au *et al.*'s method after mesh reconstruction produces nearly-identical results to applying our method directly on the point cloud model (first row of Fig. 6). However, if the input has missing data, our method is more likely to produce a correct skeleton, since their method strictly adheres to the topology defined by the incorrect reconstructed mesh. This is verified experimentally; see Fig. 6(b) as an example of the general trend.

Fig. 7 shows some point cloud models and curve skeletons extracted by our method. The inputs in the first row

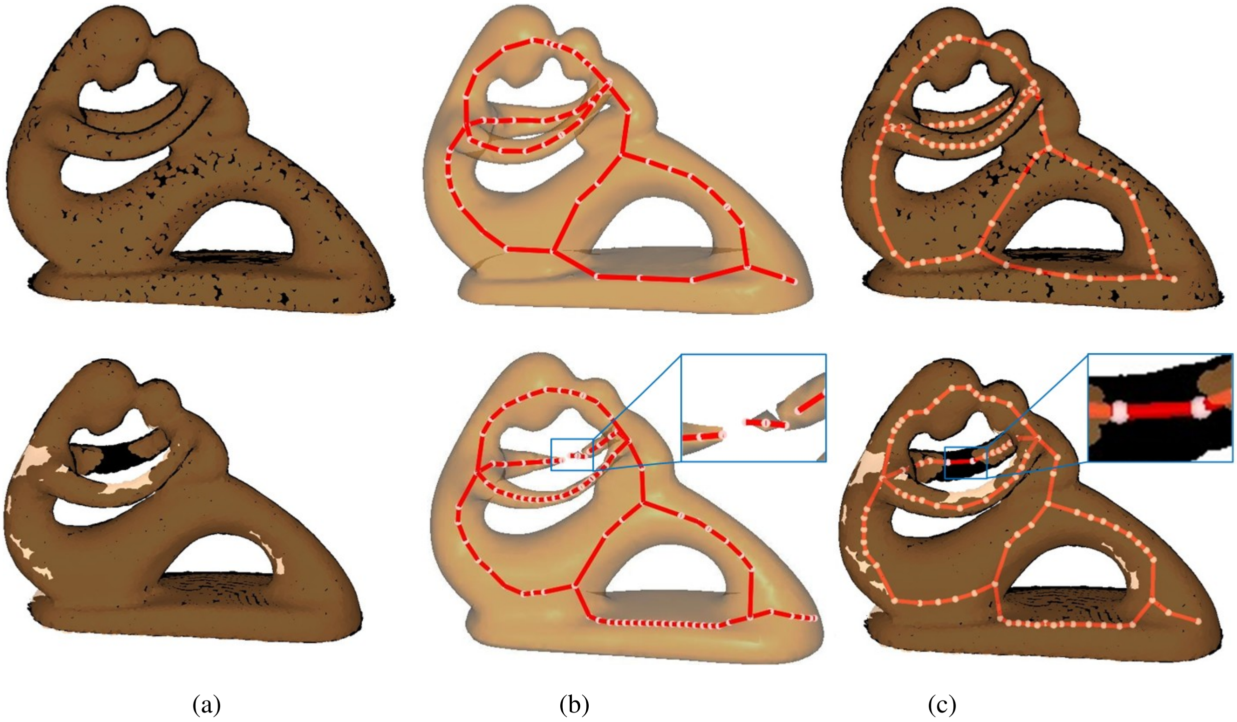


Figure 6. Au *et al.*'s method *vs.* our method. Column (a) shows the input point cloud: complete point cloud and point cloud with missing data created by a virtual scanner from four views. Column (b) shows the mesh obtained by Poisson reconstruction and the skeleton extracted using Au *et al.*'s method. Column (c) shows the skeleton extracted by our method.

are all triangular mesh models with connectivity discarded. We can properly identify the curve skeleton in sphere-like regions, as in Fig. 7(a), and sheet-like regions, as in Fig. 7(b). Note that even without any connectivity input, we can distinguish some close-by structures, for example the claws in Fig. 7(b), (c), and (d). This mainly depends upon the tangent-plane Delaunay triangulation.

The incomplete point clouds in Fig. 7(e) and (f) are composed of points captured from one (resp. four) views of a complete surface model using a virtual scanner. We generate an unnecessary cycle near the missing arm of Fig. 7(e) due to the local point distribution, but find a correct skeleton with moderate portions of missing data in Fig. 7(f). Models (g) and (h) are raw-scanned data with large missing parts. We produce a correct skeleton for Fig. 7(g) even though the model contains noise and outliers. The skeleton in Fig. 7(h) is also largely correct despite an additional cycle around the belly area.

Our method also works well for surfaces with boundaries as shown in Fig. 7(i) and (j). The two models are raw scan data for a leaf and corn, respectively, with non-uniform point distribution. Au *et al.*'s mesh-based implementation [4] failed to complete the contraction process for reconstructed meshes from the two inputs. Although we do not require connectivity or normals in the input, our method can preserve genus if the corresponding holes are sufficiently large. In addition to Fig. 4, we show three of such examples in Fig. 7(f), (k), and (l).

Our method is robust against changes in sampling resolution, as shown for example in Fig. 8, since the initial contraction weights are sampling aware. In addition to this, we can handle non-uniform distribution of samples (left example in Fig. 8). There is no obvious difference between the two extracted skeletons.

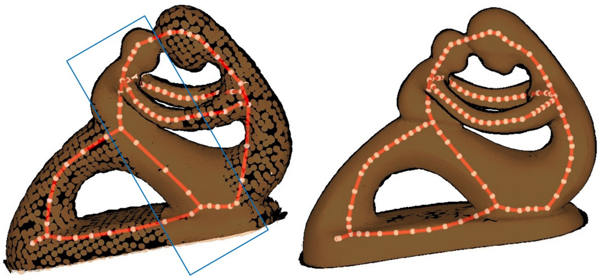


Figure 8. Our method is largely insensitive to nonuniform sampling. The left model contains 9K points, with 50% concentrated in the marked region. The right model contains 25K points with a nearly uniform distribution. The extracted curve skeletons for different resolutions and distributions are similar.

Our method is also robust against real scanner noise and outliers, as shown in Fig. 7(g). Note that Fig. 7(e) also shows that the outlier near the thigh does not affect the shape of skeleton branches around it. Another form of noise found in raw scanned data is caused by mis-alignment of several scans from different views. To simulate this case, we conduct five virtual scans from different directions along the coordinate axes, register them perfectly (as we are using a virtual scan), and then apply small (δ) random rotations and translations to each scan. As we can see in Fig. 9, the skeletons extracted from the model with $\delta = 0.5\%$ or 1% mis-alignments are almost identical to the skeleton from the original model in Fig. 6(c), except for some additional little branches

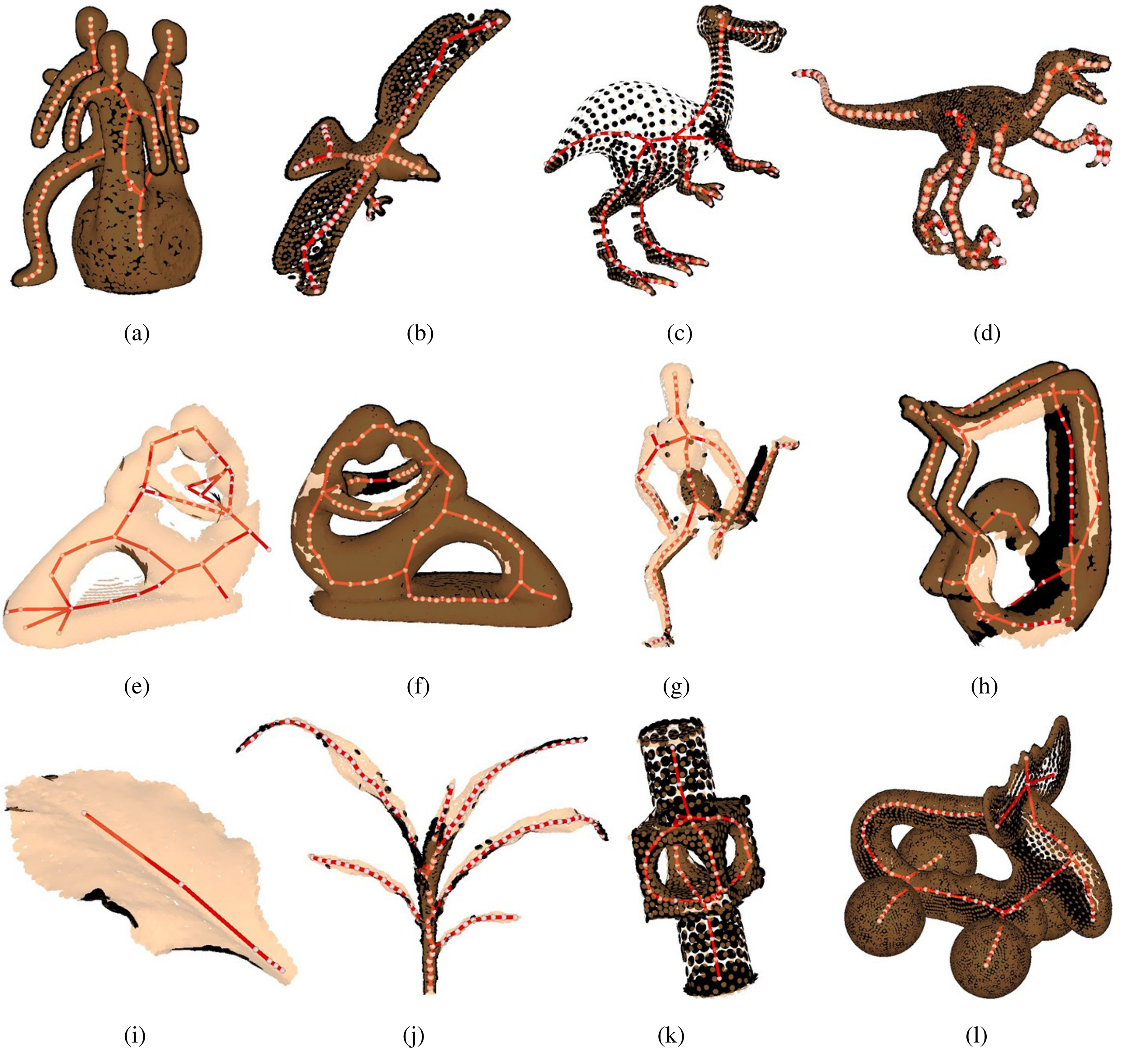


Figure 7. A gallery of our curve skeleton results. Our method extracts high quality curve skeletons for models with boundaries, flat regions, high-genus, and even with moderate amount of missing data.

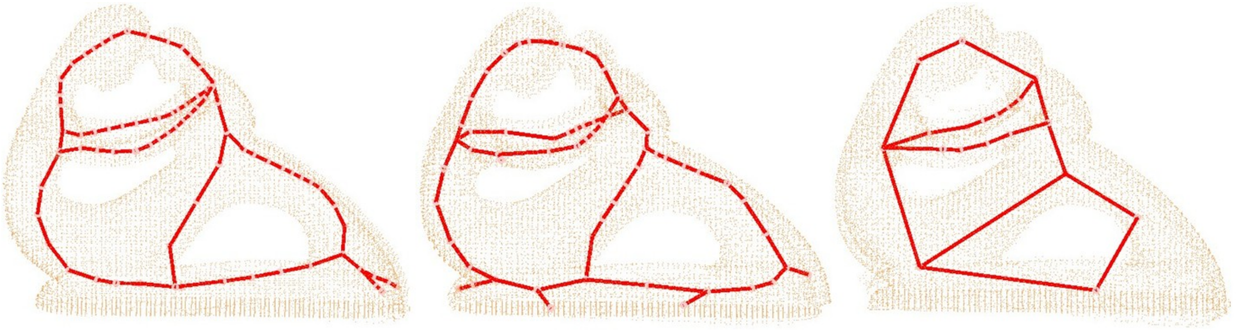


Figure 9. Our method is robust against misalignment noise. The input point cloud models are composed by mis-aligning five virtual scans by 0.5% (left), 1% (center) and 1.5% (right), respectively. See text for details of these percentages.

caused by the noise. If we increase the mis-alignment to $\delta = 1.5\%$, the shape of the skeleton changes but the result is still topologically correct.

Implementation details: The curve skeleton method is implemented in MATLAB; therefore, it is not yet optimized for speed. The most time-consuming steps are the construction of a local one-ring for each point and the iterative contraction. Our method typically takes less than 2 minutes on a point cloud with 10K points.

Each raw input cloud used in our experiments contains noise and outliers. We subsample large point cloud models to about 10K points. Hence noise and some outliers still can be found in our input; however, their presence contributes little to the final skeleton, since we solve a global system to contract the model.

B. Limitations

Close-by structures present difficulties for most point-cloud processing algorithms [51]. Since we rebuild a local approximation of connectivity from a k-nearest neighbor graph, our method is occasionally subject to topological errors when it cannot distinguish between two nearby surfaces. Our ability to extract a proper skeleton depends on the distance between samples on the manifold being smaller than the distance between the two structures. Although we are often able to extract a correct curve skeleton from those structures, like the claws in Fig. 7(b) and (c), our algorithm occasionally finds incorrect topology near those close-by structures as shown in Fig. 10(a).

Another limitation of our method is the tuning of parameters it requires. We use $s_L = 3$ for our default contraction weights update factor, which leads to convergence in few iterations for all models shown in this paper except the dancing children model in Fig. 10(b). This model has many complex interactions between thin regions and thicker parts. A lower contraction force may fail to shrink the thicker regions sufficiently, while a stronger contraction force may pull skeleton components in the thinner regions towards thicker regions. Our results in this instance are sensitive to parameter tuning. $s_L = 3$ (default) makes the contraction force increase too quickly, but $s_L = 2$ works well for the model, as in Fig. 10(c), at the cost of an extra iteration before convergence.

V. Skeleton-driven point cloud reconstruction

Given the medial axis of a surface, it is possible to precisely reconstruct the surface itself. We prefer not to work with the medial complex, which is often difficult to model and interact with, but we take advantage of the fact that a shape's curve skeleton is closely related to this complex [3]. Briefly, our technique attempts to use the curve skeleton as a sampling of the medial axis and recovers the full medial axis by interpolation.

In order to avoid dealing directly with the complex sheets forming the medial axis, we follow the solution presented in [27], where the connection between the signed distance field of the surface and the medial axis is exploited. In this setting, we strive to compute the signed distance field of the shape, so that its surface can be extracted by zero-level-set iso-surfacing.

Unfortunately, in many situations, the skeleton extracted automatically from the cloud is unsatisfactory both in terms of geometry and topology. Nevertheless, thanks to their compactness, curve skeletons are an efficient structure for topology and geometry editing. Thus, we allow the user to refine the location of skeletal nodes and modify the topology of the graph. An example of this editing process is shown in Fig. 14(a) where the skeleton topology and geometry is easily edited by the user to obtain the refined skeleton of Fig. 14(b).

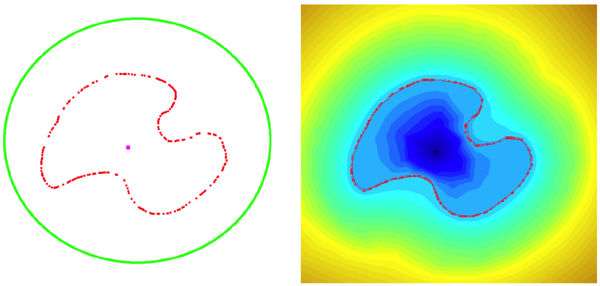


Figure 11. A 2D example of the reconstruction process on a skeleton cross-section. (a) The input data set of the cross-section; surface samples are displayed in red, the magenta dot represents the skeleton and the green circle shows the (weak) outside conditions imposed by the skeleton. (b) The iso-surface (red line) of the FEM scalar field is able to capture geometry with concavities not captured by the ω_4 term.

Given a point cloud and a refined curve skeleton, we can express the signed distance field interpolation problem using the following FEM quadratic minimization problem:

$$\begin{aligned} u = argmin\{ \quad & \omega_1 E_{smooth}(u) + \\ & \omega_2 E_c(u, P_{surf}) + \\ & \omega_3 E_c(u, P_{skel}) + \\ & \omega_4 E_c(u, P_{out}) \quad \} \end{aligned} \tag{3}$$

where E_{smooth} forces the field to be smooth:

$$E_{smooth}(u) = \int\int_{\Omega}\int \frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} + \frac{\partial^2 u}{\partial z^2} d\omega \tag{4}$$

and the remaining three terms are expressed as quadratic point constraints:

$$E_c(u, P) = \sum_{p \in P} (u(p) - k(p))^2 \tag{5}$$

Equation 3 contains three different types of point constraints. First of all, we would like the distance field to vanish close to surface samples. Thus, for cells close to the surface we set $k(p) = 0$. The second set of constraints is created for points on the skeleton, whose samples are located inside the shape; their signed distance field value is set to be the distance d from the skeleton sample to the surface: $k(p) = -d(p)$. Unfortunately, in the presence of missing data, the closest point estimate might be unreliable; see Fig. 14(c). Thus, we allow the user to visualize a cylindrical approximation and easily tune these estimates, as illustrated in Fig. 14(d). The last set of constraints is added as a regularizer, to ensure that the field will vary from a negative value inside the shape to a positive value outside the shape. We consider a position p to be outside the shape if it is far enough from any point on the skeleton (a distance of at least K times the skeleton radius, $K = 4$ in our experiments). We show fields satisfying these constraints in Fig. 11 and Fig. 12.

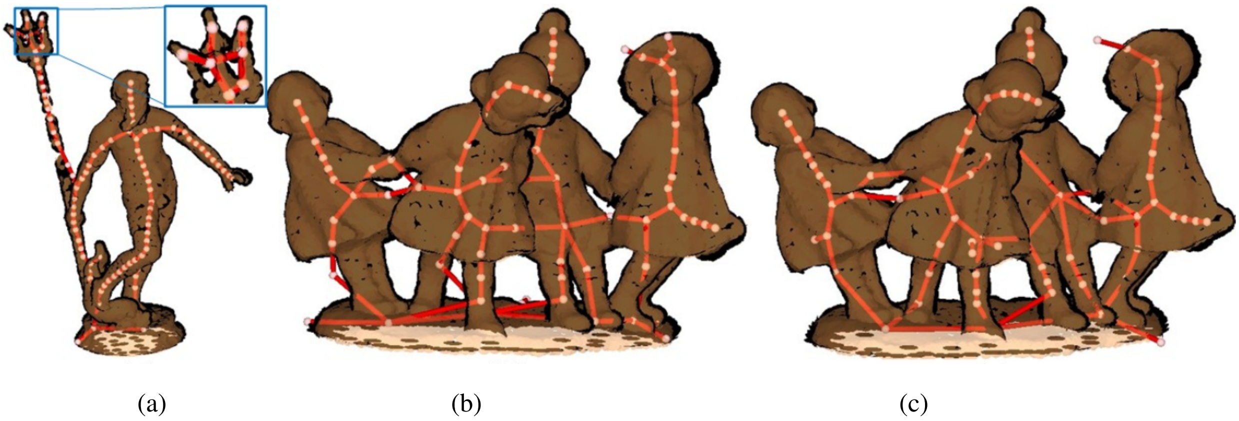


Figure 10. Two limitations of our method. (a) Incorrect topology may be produced near close-by structures, shown in the blue frame. (b) The default update factor $s_L = 3$ for contraction weights may lead to excessive contraction if there are many thin structures between thicker regions. (c) $s_L = 2$ produces better results for such case.

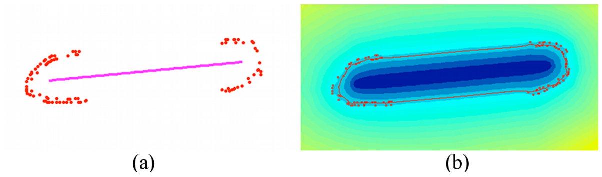


Figure 12. A 2D example of the reconstruction process along a skeleton branch. (a) The input data set. The central portion of the data is missing; surface samples are displayed in red, while the magenta curve represents the skeleton. (b) The iso-surface (red line) of the FEM scalar field is able to compensate for missing data exploiting the skeletal information.

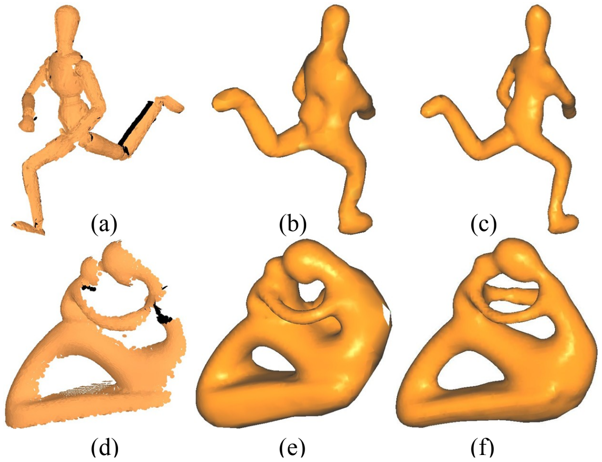


Figure 13. Reconstruction from sparse data. Left: The input point clouds. Middle: Reconstruction achieved by straightforward poisson reconstruction is under-constrained in under-sampled regions. Right: The skeleton provides topological and geometrical hints that guides reconstruction toward a more suitable solution.

We set the weights $\omega = [1,\ 5,\ .5,\ .005]$ in Equation (3) so that the field will be smooth. These weights encourage the algorithm to strongly respect both the vanishing constraints and the skeleton guesses while only loosely respecting the outside constraints.

In order to illustrate the power of this method we consider two examples. First, we consider the mesh reconstructed by Poisson on the mannequin data-set in Fig. 13(a,b); note that the surface in under-sampled regions is badly conditioned and does not accurately approximate the original surface. Here, the edited skeleton of Fig. 14(d) provides a weak prior which the FEM scheme exploits in order to create a more plausible surface in Fig. 13(c). Similarly, under-sampled regions can cause topological inconsistencies in the reconstruction of Fig. 13(d). Instead of correcting the surface, the user can easily correct the topology of the skeleton, resulting in a successful reconstruction; see Fig. 1(d,e) and Fig. 13(e,f).

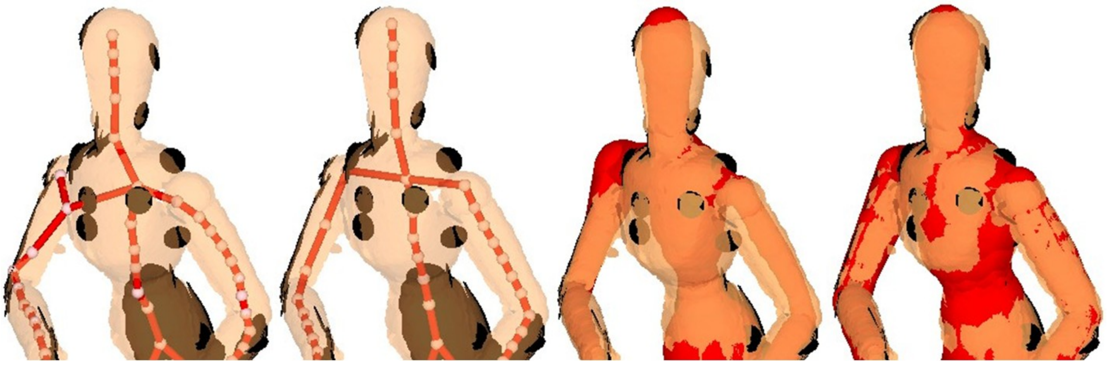
Our current implementation solves the quadratic problem on a regular spatial grid; thus, for computational reasons, the resolution of our results is limited (6 splits on the largest bounding box side). However, our preliminary results are encouraging and suggest that higher quality reconstruction could be achieved by constructing an octree adaptive domain in the spirit of [27].

VI. Conclusions and Future Work

Curve skeletons are important 1D representations of 3D models, and permit powerful and intuitive manipulations of the underlying surfaces. Although there is a great deal of work on curve skeleton extraction from volumetric and mesh representations, little attention has been paid to direct extraction from point clouds, and almost none from incomplete data.

In this paper, we present an algorithm based on Laplacian contraction. Our method is robust to noise and sample distribution, and can handle moderate amount of missing data. Our algorithm is also able to handle surfaces with boundaries and polygon soups. We show experimental results that demonstrate the effectiveness of our algorithm under a wide range of data sets. As an easy-to-maintain representation for topology repair, we also show that the extracted skeletons can assist in mesh reconstruction from incomplete point cloud on which direct application of typical reconstruction methods often fail.

In the future, we would like to improve our neighborhood construction to handle close-by structures, such as the trident in Fig. 10(a). Constructing a Mahalanobis neighborhood using the normal vector at each point is a potential solution. We will continue to explore how to use the curve skeleton to repair the point clouds directly.



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

Figure 14. Work-flow of our editing process. (a) The skeleton extracted by our contraction algorithm. (b) The skeleton edited by location correction and simple topological operations. (c) The radius field defined on the skeleton depicted as a conical shape approximation. (d) The radius field is interactively modified to better represent the shape.

ACKNOWLEDGMENT

The authors are grateful to the reviewers for their insightful and helpful comments. Geometry models used in the paper are obtained from the AIM@SHAPE shape repository, the Stanford 3D Scanning repository, Lior Shapira, the Princeton Shape Benchmark, Hugues Hoppe, the Digital Plant Laboratory, and our own laser scanning using a Polhemus Cobra FastScan System. This work is supported in part by a grant from NSERC (No. 611370) and NSFC (No. 60673006 and No. U0935004).

REFERENCES

[1] X. Li, T. W. Woon, T. S. Tan, and Z. Huang, “Decomposing polygon meshes for interactive applications,” in *Proc. of Symp. on Interactive 3D graphics*, 2001, pp. 35–42.

[2] M. Mortara, G. Patanè, M. Spagnuolo, B. Falcidieno, and J. Rossignac, “Plumber: a method for a multi-scale decomposition of 3D shapes into tubular primitives and bodies,” in *Proc. of ACM Symp. on Solid Modeling and App.*, 2004, pp. 339–344.

[3] T. K. Dey and J. Sun, “Defining and computing curve-skeletons with medial geodesic function,” in *Symp. on Geom. Proc.*, 2006, pp. 143–152.

[4] O. K.-C. Au, C.-L. Tai, H.-K. Chu, D. Cohen-Or, and T.-Y. Lee, “Skeleton extraction by mesh contraction,” *ACM Trans. on Graph*, vol. 27, no. 3, pp. 44:1–44:10, 2008.

[5] S. Biasotti, S. Marini, M. Mortara, G. Patanè, M. Spagnuolo, and B. Falcidieno, “3D shape matching through topological structures,” in *Discrete Geometry for Computer Imagery*, 2003, pp. 194–203.

[6] H. Sundar, D. Silver, N. Gagvani, and S. Dickinson, “Skeleton based shape matching and retrieval,” in *Proc. IEEE Int. Conf. on Shape Modeling and Applications*, 2003, p. 130.

[7] N. D. Cornea, M. F. Demirci, D. Silver, A. Shokoufandeh, S. J. Dickinson, and P. B. Kantor, “3D object retrieval using many-to-many matching of curve skeletons,” in *Proc. IEEE Int. Conf. on Shape Modeling and Applications*, 2005, pp. 368–373.

[8] R. Raab, C. Gotsman, and A. Sheffer, “Virtual woodwork: making toys from geometric models,” *Int. J. Shape Modeling*, vol. 10, no. 1, pp. 1–30, 2004.

[9] A. Tagliasacchi, H. Zhang, and D. Cohen-Or, “Curve skeleton extraction from incomplete point cloud,” *ACM Trans. on Graph*, vol. 28, no. 3, 2009.

[10] L. Wade and R. E. Parent, “Automated generation of control skeletons for use in animation,” *Visual Computer*, vol. 18, no. 2, pp. 97–110, APR 2002.

[11] F.-C. Wu, W.-C. Ma, R.-H. Liang, B.-Y. Chen, and M. Ouhyoung, “Domain connected graph: the skeleton of a closed 3D shape for animation,” *Visual Computer*, vol. 22, no. 2, pp. 117–135, 2006.

[12] I. Baran and J. Popović, “Automatic rigging and animation of 3D characters,” *ACM Trans. on Graph*, vol. 26, no. 3, p. 72, 2007.

[13] H.-B. Yan, S. Hu, R. R. Martin, and Y.-L. Yang, “Shape deformation using a skeleton to drive simplex transformations,” *IEEE Trans. Vis. & Comp. Graphics*, vol. 14, no. 3, pp. 693–706, 2008.

[14] E. de Aguiar, C. Theobalt, S. Thrun, and H.-P. Seidel, “Automatic conversion of mesh animations into skeleton-based animations,” *Computer Graphics Forum*, vol. 27, no. 2, pp. 389–397, APR 2008.

[15] K. Siddiqi and S. Pizer, *Medial Representations: Mathematics, Algorithms and Applications*. Springer, 2008.

[16] M. Hilaga, Y. Shinagawa, T. Kohmura, and T. L. Kunii, “Topology matching for fully automatic similarity estimation of 3D shapes,” in *Proc. of SIGGRAPH*, 2001, pp. 203–212.

[17] J. Chuang, N. Ahuja, C. Lin, C. Tsai, and C. Chen, “A potential-based generalized cylinder representation,” *Computers & Graphics*, vol. 28, no. 6, pp. 907–918, 2004.

[18] P. Shilane, P. Min, M. Kazhdan, and T. Funkhouser, “The princeton shape benchmark,” in *Proc. IEEE Int. Conf. on Shape Modeling and Applications*, 2004.

[19] Y. Ohtake, A. G. Belyaev, M. Alexa, G. Turk, and H.-P. Seidel, “Multi-level partition of unity implicits,” *ACM Trans. on Graph*, vol. 22, no. 3, pp. 463–470, 2003.

[20] O. Schall, A. G. Belyaev, and H.-P. Seidel, "Error-guided adaptive fourier-based surface reconstruction," *Computer-Aided Design*, vol. 39, no. 5, pp. 421–426, 2007.

[21] M. Kazhdan, M. Bolitho, and H. Hoppe, "Poisson surface reconstruction," in *Symp. on Geom. Proc.*, 2006, p. 70.

[22] A. Jalba and J. B. T. M. Roerdink, "Efficient surface reconstruction using generalized coulomb potentials," *IEEE Trans. Vis. & Comp. Graphics*, vol. 13, no. 6, pp. 1512–1519, 2007.

[23] Y. Pekelny and C. Gotsman, "Articulated object reconstruction and markerless motion capture from depth video," *Computer Graphics Forum (Special Issue of Eurographics)*, vol. 27, no. 2, pp. 399–408, 2008.

[24] W. Chang and M. Zwicker, "Range scan registration using reduced deformable models," *Computer Graphics Forum (Special Issue of Eurographics)*, vol. 28, no. 2, pp. 447–456, 2009.

[25] M. Wand, B. Adams, M. Ovsjanikov, A. Berner, M. Bokeloh, P. Jenke, L. Guibas, H.-P. Seidel, and A. Schilling, "Efficient reconstruction of non-rigid shape and motion from real-time 3D scanner data," *ACM Trans. on Graph*, vol. 28, no. 2, 2009.

[26] Q. Zheng, A. Sharf, A. Tagliasacchi, B. Chen, H. Zhang, A. Sheffer, and D. Cohen-Or, "Consensus skeleton for non-rigid space-time registration," *Computer Graphics Forum (Special Issue of Eurographics)*, p. to appear, 2010.

[27] A. Sharf, T. Lewiner, G. Shklarski, S. Toledo, and D. Cohen-Or, "Interactive topology-aware surface reconstruction," *ACM Trans. on Graph*, vol. 26, no. 3, p. 43, 2007.

[28] Q.-Y. Zhou, T. Ju, and S.-M. Hu, "Topology repair of solid models using skeletons," *IEEE Trans. Vis. & Comp. Graphics*, vol. 13, no. 4, pp. 675–685, 2007.

[29] T. Ju, Q.-Y. Zhou, and S.-M. Hu, "Editing the topology of 3D models by sketching," *ACM Trans. on Graph*, vol. 26, no. 3, p. 42, 2007.

[30] N. D. Cornea, D. Silver, and P. Min, "Curve-skeleton properties, applications, and algorithms," *IEEE Trans. Vis. & Comp. Graphics*, vol. 13, no. 3, pp. 530–548, 2007.

[31] S. Fang, S. Fang, H. Chen, and H. Chen, "Hardware accelerated voxelization," *Computers and Graphics*, vol. 24, no. 3, pp. 433–442, 2000.

[32] D. Morgenthaler, "Three-dimensional simple points: Serial erosion, parallel thinning and skeletonization," Computer Vision Laboratory, Computer Science Center, Univ. of Maryland, College Park, MD., Tech. Rep. 1005, 1981.

[33] S. Svensson, I. Nystrom, and G. S. di Baja, "Curve skeletonization of surface-like objects in 3D images guided by voxel classification," *Pattern Recognition Letters*, vol. 23, no. 12, pp. 1419–1426, 2002.

[34] A. Sharf, T. Lewiner, A. Shamir, and L. Kobbelt, "On-the-fly curve-skeleton computation for 3D shapes," *Computer Graphics Forum (Special Issue of Eurographics)*, vol. 26, no. 3, pp. 323–328, 2007.

[35] R. Ogniewicz and M. Ilg, "Voronoi skeletons: theory and applications," Jun 1992, pp. 63–69.

[36] F. Lazarus and A. Verroust, "Level set diagrams of polyhedral objects," in *Proc. ACM Symp. on Solid Modeling and App.*, 1999, pp. 130–140.

[37] M. Attene, S. Biasotti, and M. Spagnuolo, "Shape understanding by contour-driven retiling," *Visual Computer*, vol. 19, no. 2-3, pp. 127–138, 2003.

[38] G. Aujay, F. Hétroy, F. Lazarus, and C. Depraz, "Harmonic skeleton for realistic character animation," in *Proc. of Eurographics Symp. on Computer Animation*, 2007, pp. 151–160.

[39] A. Verroust and F. Lazarus, "Extracting skeletal curves from 3D scattered data," *Visual Computer*, vol. 16, no. 1, pp. 15–25, 2000.

[40] G. Taubin, "A signal processing approach to fair surface design," in *Proc. of SIGGRAPH*. New York, NY, USA: ACM, 1995, pp. 351–358.

[41] D. Field, "Laplacian smoothing and Delaunay triangulations," *Communications in applied numerical methods*, vol. 4, no. 6, pp. 709–712, 1988.

[42] N. Amenta, S. Choi, and R. K. Kolluri, "The power crust," in *Proc. of ACM Symp. on Solid Modeling and App.*, 2001, pp. 249–266.

[43] J. Carr, R. Beatson, J. Cherrie, T. Mitchell, W. Fright, B. McCallum, and T. Evans, "Reconstruction and representation of 3D objects with radial basis functions," in *Proc. of SIGGRAPH*. ACM, 2001, p. 76.

[44] O. Sorkine and D. Cohen-Or, "Least-squares meshes," in *Proc. IEEE Int. Conf. on Shape Modeling and Applications*, 2004, pp. 191–199.

[45] C. Luo, I. Safa, and Y. Wang, "Approximating gradients for meshes and point clouds via diffusion metric," *Computer Graphics Forum*, vol. 28, no. 5, pp. 1497–1508, 2009.

[46] M. Belkin, J. Sun, and Y. Wang, "Constructing laplace operator from point clouds in r^d," in *Proc. of ACM Symp. on Discrete Algorithms*, 2009, pp. 1031–1040.

[47] M. S. Floater, "Mean value coordinates," *Computer Aided Geometric Design*, vol. 20, no. 1, pp. 19–27, 2003.

[48] M. Meyer, M. Desbrun, P. Schröder, and A. H. Barr, "Discrete differential-geometry operators for triangulated 2-manifolds," in *Visualization and Mathematics III*, H.-C. Hege and K. Polthier, Eds., 2003, pp. 35–57.

[49] L. Shapira, A. Shamir, and D. Cohen-Or, "Consistent mesh partitioning and skeletonization using the shape diameter function," *Visual Computer*, vol. 24, no. 4, pp. 249–259, 2008.

[50] T. Dey and J. Sun, "Normal and feature approximations from noisy point clouds," *Lecture Notes in Computer Science*, vol. 4337, p. 21, 2006.

[51] H. Huang, D. Li, H. Zhang, U. Ascher, and D. Cohen-Or, "Consolidation of unorganized point clouds for surface reconstruction," *ACM Trans. on Graph*, vol. 28, no. 5, p. Article 176, 2009.