www.cadanda.com Computer-Aided Design AND APPLICATIONS

A Robust and Centered Curve Skeleton Extraction from 3D Point Cloud

Vanna Sam[1], Hiroaki Kawata[2] and Takashi Kanai[3]

[1]The University of Tokyo, sam_vanna@graco.c.u-tokyo.ac.jp
[2]The University of Tokyo, hiroaki@graco.c.u-tokyo.ac.jp
[3]The University of Tokyo, kanai@graco.c.u-tokyo.ac.jp

ABSTRACT

A curve skeleton of a 3D object is one of the most important structures of the object, which is extremely useful for many computer graphics applications involving in shape analysis. Much research has focused on volumetric and polygonal mesh models. However, only a few have been paid attention to point cloud and suffered from several limitations such as connectivity or inside-shape information requirements, difficulty to compute junction nodes, or inappropriate parameter tuning. To resolve these problems, we propose an effective algorithm for extracting a robust and centered curve skeleton from point cloud. The process starts from estimating centers of antipodes of a point cloud, so called skeletal candidates, which are fundamentally inside the shape; but scattered. We thus filter and shrink them to create less-noisy skeletal candidates before applying one-dimensional Moving Least Squares (MLS) to build a thin point cloud. The latter is then downsampled to sparse skeletal nodes. With recent nodes, it is possible to create a smooth curve skeleton; but distorted due to shrinking and thinning (MLS) processes. We thus utilize cross-section plane technique and least squares ellipse fitting to relocate skeletal nodes to create a centered curve skeleton. The algorithm is validated on many complex models ranging from cylindrical to planar ones, and from clean to noisy and incomplete data. We show that the extracted curve skeleton is robust to noises and under centeredness property.

Keywords: curve skeleton, point cloud, antipode, regression line, ellipse fitting
DOI: 10.3722/cadaps.2012.869-879

1 INTRODUCTION

A curve skeleton of 3D complex shape is a compact and simple structure basically called 1D curve which provides sufficient information on geometry and topology of the shape. There is no absolute definition of a curve skeleton and several have been proposed in the literature review [7]. It is defined as the locus of the centers of maximal inscribed balls [4], the centerline of 3D objects [12], or as the rotational symmetry axis of an object [24]. The most commonly-targeted curve skeleton is a one-dimensional poly-segment connecting central points of equidistance to the nearest boundary of the shape. Due to its simple structure, a curve skeleton is easy to manipulate and very useful in many computer graphics applications which involve in shape analysis such as shape recognition, object matching and retrieval [6], animation transfer, and so forth. These applications correspondingly require several necessary properties among topology preservation, centeredness, robustness to noises, connectedness (as a single graph), reliability (which guarantees that at least one boundary point is visible to the curve skeleton), hierarchy and reconstructability [7].

The major problem in extracting a curve skeleton is how to compute the shape correctly and robustly. Even though numerous algorithms have been developed in the past and produced remarkable results, to our knowledge, much research has been done on only polygonal meshes and volumetric models [2],[17],[23], which require full knowledge of models. Some have been also paid attention to both mesh and point cloud models [12],[20],[21]. However, only a few have been focusing on noisy and incomplete data models [5],[24]. The challenge to directly extract a curve skeleton from noisy or incomplete data is still on-going because previous research suffers from several limitations such as÷ models must be meshed or watertight [2], curve skeletons at junctions are unreliable and hard to compute [20],[24] and a lot of efforts are needed in parameter tuning [5],[24].

To overcome these problems, we propose a robust extraction of curve skeleton from clean, noisy and incomplete data. There are two key ideas in our method. One is to make our extraction algorithm robust: We validate the skeletal candidates according to their densities based on filtering and shrinking processes. Low density regions of skeletal candidates contribute less in creating a curve skeleton. They are, thus, considered as noises or outliers and are eliminated by filtering process. The remaining skeletal candidates are reinforced through shrinking process to create ones less noises and free from outliers. The other is that we utilize ellipse fitting to relocate the skeletal nodes to ensure that they lie on the center of the model *i.e.* our curve skeleton is under robustness and centeredness properties.

Our main contributions include (i) a curve skeleton extraction directly from point cloud (ii) a robust extraction from noisy and incomplete data and (iii) a centeredness-guaranteed curve skeleton defined by ellipse fitting.

The paper is organized as follows. Section 2 presents literature review of curve skeleton extraction methods and related work. In Section 3, we describe details of our proposed algorithm, and we show results and discussion of our algorithm on various types of 3D shapes in Section 4. At the end of the section, we show a comparison of our algorithm with [5] and [24]. Section 5 raises conclusions and future work for our method.

2 RELATED WORK

There is a vast literature of curve skeleton starting with the most well-known medial axis [4]. There are numerous algorithms in extracting curve skeletons in the past. Those algorithms are roughly categorized into two main groups. For a complete review, we refer readers to [7] and [22].

Volumetric methods: In this category, a given 3D input model is voxelized in 3D space and the curve skeleton is computed by volumetric thinning or by field-function techniques. In volumetric thinning [23],[17],[3], boundary voxels so called simple points are iteratively removed until one voxel thick is found. The curve skeleton in this method is a result of connecting centers of central voxels. Whereas in field-function techniques [6],[12], distance or force field is defined for each interior point and the ridges of this field correspond to central voxels of the object. The curve skeleton results from connecting those ridges. In summary, this type of methods requires clear information on the inside of the shape and suffers numerical instability caused by inappropriate voxelization resolutions clearly mentioned in [5],[24]. It is, thus, not suitable for point cloud models.

Geometric Methods: This kind of methods is applicable to both polygonal meshes and point clouds. The most popular techniques involve in using Voronoi diagram [8],[18],[19] to approximate medial surfaces and then prune them to create a curve skeleton. Recent research [2] utilizes Laplacian smoothing to shrink polygonal mesh to zero volume mesh and then fix the connectivity to create a curve skeleton. This method provides excellent results. However, it is applicable to only polygonal mesh and water-tight models. [5] can be considered as an extension of the previous method for incomplete point clouds. Each point requires a local creation of one-ring connectivity of neighborhood for Laplacian operation and much attention has to be paid to contraction parameters. The research produces good results for some models. However, incorrect topology may occur near close-by structures and wrong parameter tuning may lead to a curve skeleton outside the model. Recently, [24] utilizes cutting plane idea to compute a curve skeleton from incomplete point cloud and fills in the missing data to assist surface reconstruction. Cutting planes are well oriented at cylindrical shapes; but not at non-cylindrical ones, which are normally junction regions. The latter requires special handling. However, it seems hard to define and guarantee that curve skeletons around junctions are well connected or centered.

Similar to [20], we attempt to approximate antipodal points from the shape and their corresponding centers called skeletal candidates for a curve skeleton. Different from [20], we send only one ray to find the antipode of each point and we do not eliminate skeletal candidates at junction regions. The latter will be well treated in Section 3 and the centeredness is verified at the end of the section.

3 CURVE SKELETON EXTRACTION

Point-based models are usually acquired by range scanner or image-based reconstruction techniques. They are ready to use, easy to handle, and do not require any connectivity information or complex data structures such as polygonal mesh models. Our proposed algorithm focuses on oriented point clouds to make use of the above advantages. Given a point cloud as an input, the normal direction and orientation of a point can be estimated through its neighborhood information [13],[14].

Our algorithm proceeds as follows. We first attempt to compute the center of antipodes of each point in the cloud by extending its normal in reverse direction till the opposite side of the boundary. All computed centers are called *skeletal candidates*. The latter are basically scattered and theoretically lain on medial surface. To conquer the weakness of medial axis, which is sensitive to noisy boundary surface, we assume skeletal candidates of high density neighborhood contribute better to the computation of a curve skeleton in the next process. We thus consider skeletal candidates of low density as outliers or noises. It means that we remove them and shrink the remainder by Laplacian smoothing [25] to form a strong-bond distribution. We then process thinning algorithm using 1D Moving Least Squares [15],[16] till a line-like thin cloud of skeletal candidates is obtained. The next process is to extract skeletal nodes necessary for a curve skeleton. The final process is to relocate those skeletal nodes to the center of the shape. Below are details of each process.

3.1 Antipodes and Skeletal Candidates

Two points are antipodes if they are diametrically opposite. On a smooth surface, an antipodal point can defined by extending its normal in reverse direction till the opposite boundary surface [20]. However, in polygonal meshes or point clouds, exact antipodal points are difficult to compute. In [20], they approximate an antipodal point by sending out several rays per point in an open angle of 120 degree to the opposite boundary of polygonal mesh. On the other hand, in our case, we do not see necessity in sending out so many rays because the noisy computed centers of antipodes will be well handled in next processes.

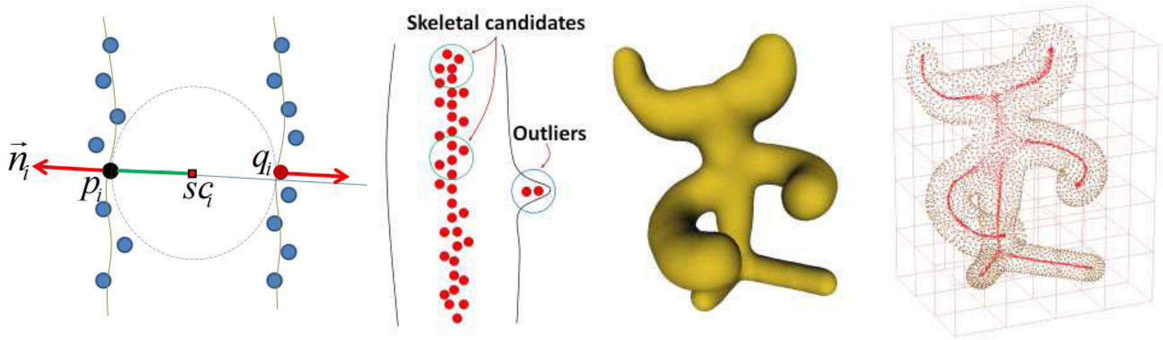


Fig. 1: (a) Approximate antipode and its center. (b) Skeletal candidates with different density. (c) Input model. (d) Gridding and noisy skeletal candidates on a model.

Let $(\mathbf{p}_i, \mathbf{n}_i) \in \mathbf{P}$ be a point and its normal vector in the cloud $\mathbf{P}$. We first send a ray from $\mathbf{p}_i$ in reverse- normal direction till we find a point on the opposite side of the boundary called an antipodal point $\mathbf{q}_i$ (See Fig. 1(a)). In the perfect case, the normal of $\mathbf{q}_i$ and $\mathbf{n}_i$ lie on the same direction; but in real case, it is hard to find. We thus adopt $\mathbf{q}_i$ as an antipode if and only if it respects three conditions. One is that its normal forms an obtuse angle with $\mathbf{n}_i$, second is that it lies close to $\mathbf{n}_i$, and third is that $\mathbf{q}_i$ provides the smallest distance with $\mathbf{p}_i$. The center of the antipodes, s*keletal candidate* $\mathbf{sc}_i$, is located at a half of the smallest distance and lies on the normal direction $\mathbf{n}_i$. Skeletal candidate $\mathbf{sc}_i$ is under two constraints:

- It must locate inside the object
- It is at high density region (Fig. 1(b))

The computation can be sped up by gridding the whole point cloud into blocks (Fig. 1(d)). Each computation of an antipode utilizes only several blocks through which reverse- direction ray runs.

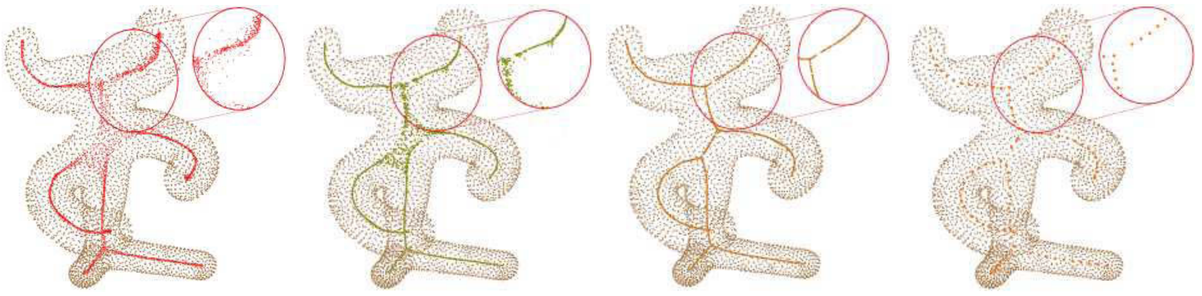
3.2 Filtering and Thinning Process

The problem here is identical to curve reconstruction from an unorganized point cloud. Skeletal candidates are in general noisy and are not ready for curve skeleton extraction. To create a curve skeleton, we need a line- like distribution of point samples. The latter can be obtained by projecting $\mathbf{sc}_i$ onto their regression lines using Moving Least Squares (MLS) technique [15],[16]. Beforehand we need a clean distribution of skeletal candidates. We thus eliminate $\mathbf{sc}_i$ of low density neighborhood $\rho_i < \rho$ and shrink them to create less noisy skeletal candidates with Laplacian smoothing (Fig. 2(b)). A prescribed density rate ρ and the feature size θ_{shrink} are described in Section 4.

MLS method is a well- known technique to make scattered points as thin as possible with respect to original shape. The basic idea of MLS is to project each point $\mathbf{sc}_i$ onto its regression curve $\boldsymbol{\pi}_i$ estimated within neighborhood N_i of a local feature size θ_{mls}. Then $\mathbf{sc}_i$ is moved to the new location on $\boldsymbol{\pi}_i$. This process can be done by computing a regression plane $\boldsymbol{\pi}_i = ax + by + c$ by minimizing the quadratic function D_i:

$$D_i = \sum_{p_j \in N_i} (ax_j + by_j + c)^2 w_j,$$

where w_j denotes a weight function. We can then solve MLS problem in 2D *i.e.* line fitting problem to 2D points. In our case, we intend to compute discrete curve skeleton, we thus do not need to compute 3D regression curve; but 3D regression line. We simply compute 3D regression line by using the Principle Component Analysis (PCA). The line passes through the centroid of PCA and directs along the eigenvector of the largest eigenvalue of the covariance matrix of PCA. We iteratively project each point $\mathbf{sc_i}$ onto its 3D regression line until 1D point cloud is obtained (Fig. 2(c)). The process is repeated four times through our experiment and results are sufficient enough for skeletal nodes' sampling.



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

Fig. 2: Curve skeleton extraction process. (a) Skeletal candidates. (b) Filtered and skeletal candidates. (c) MLS projected skeletal candidates. (d) Pre- centered skeletal nodes.

3.3 Skeletal Node Sampling

The previous processes provide projected skeletal candidates sufficiently thin and dense enough for building 1D curve skeleton. We uniformly resample projected skeletal candidates in a large reduction scale [15] (Fig. 2(d)). We first pick a projected skeletal candidate randomly from a thin cloud as a skeletal node $\mathbf{sn}_k$ and compute its neighboring points N_k within a local feature size θ_{sample} using KD tree data structure. We then choose the furthest point in N_k from $\mathbf{sn}_k$, and define it as another skeletal node $\mathbf{sn}_{k+1}$. The remainder in N_k is removed. The process is repeated until no point is left in the stack. Finally we obtain several dozen of points as shown in Fig. 2(d).

3.4 Centering Process and Curve Extraction

Skeletal nodes obtained from the previous processes can be used to create a smooth curve skeleton. However, shrinking and thinning processes may distort the centeredness of computed skeletal nodes. We thus utilize least- squares ellipse fitting in [10],[11] to relocate skeletal nodes. Ellipse fitting process has to be done in 2D space. We thus need to extract relevant neighborhood N_k of each $\mathbf{sn}_k$ and transform N_k from 3D to 2D space before applying ellipse fitting algorithm. We detail the whole process into three steps as follows and the result is shown in Fig. 3(b):

Step 1 (Relevant Neighborhood N_k): We utilize the cutting plane idea introduced in [24] to collect boundary points within the thickness δ_{plane} to the plane. The latter passes through the skeletal node $\mathbf{sn}_k$ and its orientation is a vector built from a node $\mathbf{sn}_k$ and its nearest neighbor. The cutting plane may pass through multiple branches of the model. To extract relevant neighboring points from the latter, Density- Based Spatial Clustering of Application with Noise (DBSCAN) [9] is a suitable candidate since we have no information on number of clusters (Fig. 3(a)). The targeted cluster is one whose centroid is closed to $\mathbf{sn}_k$. Two necessary parameters (eps, MinPts) for DBSCAN are described in Section 4.

Step 2 (Basis Matrix Estimation): Once relevant neighboring points N_k are found, we transform them from 3D into 2D coordinate system for ellipse fitting process. To estimate an optimized basis matrix for 3D projection, we make use of cutting plane optimization problem introduced in [24] such that the cutting plane orientation is optimized when it is perpendicular to the tangent plane formed from points in N_k. The problem can be solved in closed-form by computing three eigenvectors of covariance matrix $\mathbf{M}_{3x3}$ of normal vectors of points in N_k (more detailed in [24]),

$$\mathbf{M}_{3x3} = \begin{pmatrix} \overline{X^2} - \overline{X}^2 & 2\overline{XY} - 2\overline{X}\,\overline{Y} & 2\overline{XZ} - 2\overline{X}\,\overline{Z} \\ 2\overline{XY} - 2\overline{X}\,\overline{Y} & \overline{Y^2} - \overline{Y}^2 & 2\overline{YZ} - 2\overline{Y}\,\overline{Z} \\ 2\overline{XZ} - 2\overline{X}\,\overline{Z} & 2\overline{YZ} - 2\overline{Y}\,\overline{Z} & \overline{Z^2} - \overline{Z}^2 \end{pmatrix},$$

where X, Y, Z and $\overline{X}, \overline{Y}, \overline{Z}$ are random variables and averages of x,y,z components of normals in N_k respectively. The three eigenvectors $\boldsymbol{\vartheta}_1, \boldsymbol{\vartheta}_2, \boldsymbol{\vartheta}_3$ of $\mathbf{M}_{3x3}$ can be solved via singular value decomposition. The projection matrix is then defined as:

$$\mathbf{Pr} = \begin{pmatrix} \vartheta_{1x} & \vartheta_{1y} & \vartheta_{1z} \\ \vartheta_{2x} & \vartheta_{2y} & \vartheta_{2z} \\ \vartheta_{3x} & \vartheta_{3y} & \vartheta_{3z} \end{pmatrix}.$$

Step 3 (Least Squares Methods of Ellipse Fitting): In this step, we intend to compute all centers of ellipses fitted to a set of 2D points in N_k for every skeletal node. The center is a correct node if it locates at the center of the shape and lies close to its corresponding skeletal node i.e. the distance between the center and its skeletal node must be smaller than an error ε_{center}. Otherwise it is neglected.

An ellipse fitting is a particular case of general conic fitting. Below is a short explanation of an algebraic ellipse fitting (more details in [11]).

$$F(x, y) = ax^2 + bxy + cy^2 + dx + ey + f = 0. \tag{1}$$

An ellipse is defined by six coefficients (a, b, c, d, e, f) with a constraint $b^2 - 4ac < 0$. (x, y) are coordinates of 2D points to be fitted to an ellipse. In matrix form (1) can be written as:

$$F_{\mathbf{u}}(\mathbf{X}) = \mathbf{X} \cdot \mathbf{u} \quad \text{where } \mathbf{X} = [x^2, xy, y^2, x, y, 1] \text{ and } \mathbf{u} = [a, b, c, d, e, f]^T. \tag{2}$$

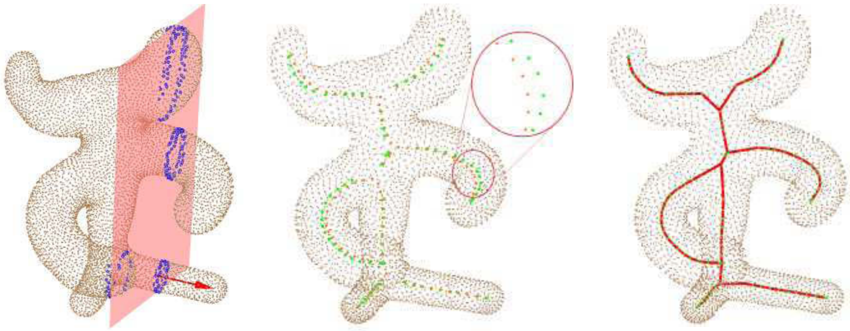
To find the six coefficients of an ellipse $\mathbf{u}$, we solve the above equation by minimizing the sum of squared algebraic distances from points to the ellipse:

$$\min_{\mathbf{u}} \sum_{j=1}^{N} F(x_j, y_j)^2 = \min_{\mathbf{u}} \sum_{j=1}^{N} F(\mathbf{X}_j \cdot \mathbf{u})^2 = \min_{\mathbf{u}} \|\mathbf{D}\mathbf{u}\|^2. \tag{3}$$

In order to simplify the computation and to obtain a proper solution, the constraint $b^2 - 4ac < 0$ is determined to $4ac - b^2 = 1$. The latter can be rewritten in matrix form $\mathbf{u}^T\mathbf{C}\mathbf{u} = 1$ and the solution of the minimization problem (3) is the eigenvector corresponding to the smallest positive eigenvalue of generalized eigenproblem.

$$\mathbf{D}^T\mathbf{D}\mathbf{u} = \lambda\mathbf{C}\mathbf{u}, \text{with } \mathbf{D} = \begin{pmatrix} x_1^2 & x_1y_1 & y_1^2 & x & y & 1 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ x_j^2 & x_jy_j & y_j^2 & x_j & y_j & 1 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ x_M^2 & x_My_M & y_M^2 & x_M & y_M & 1 \end{pmatrix}, \mathbf{C} = \begin{pmatrix} 0 & 0 & 2 & 0 & 0 & 0 \\ 0 & -1 & 0 & 0 & 0 & 0 \\ 2 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix}.$$

Curve Extraction: To construct a curve skeleton, we apply nearest neighbor crust algorithm [1] on recently centered skeletal nodes (Fig. 3(c)).

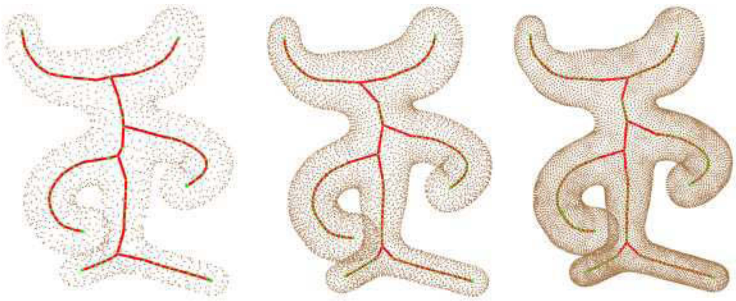


(a) (b) (c)

Fig. 3: Centering Process. (a) Cutting plane with an orientation (red vector) and relevant neighborhood. (b) Difference between pre-centered (Red) and centered skeletal nodes (Green). (c) Curve skeleton.

4 RESULTS AND DISCUSSION

We have demonstrated our algorithm on different models including clean, noisy and incomplete data. Our algorithm works well even in low resolution model (2K points) and provides not much different results with high resolution ones (Fig.4). Since there is no need to use a great number of points for a curve skeleton extraction in our algorithm, we thus downsample the tested models to approximately 10K points for efficiency.



(a) (b) (c)

Fig. 4: Consistency of our curve skeleton. (a) 2K points. (b) 8K points. (c) 15K points.

Parameters: The parameters $(\rho, \theta_{shrink}, \theta_{mls}, \theta_{sample}, \delta_{plane}, \epsilon_{center}, \text{eps}, \text{MinPts})$ can be independently controlled to extract a curve skeleton adapting to particular topology of the model. Tuning many parameters is a tedious task and it takes much effort to find good ones. In our case, we empirically notice that there is no need to tune all eight parameters; but two of them – ρ and θ_{shrink}. In this paper, we choose ρ from 6 to 18 points (a point whose neighboring points are less than ρ is considered as outlier) and θ_{shrink} is set from 1.5 to 2.5% of the diagonal of the bounding box. Other parameters are estimated as $\theta_{mls} = 2\theta_{shrink}$, $\theta_{sample} = \delta_{plane} = \epsilon_{center} = \text{eps} = \theta_{shrink}$, $\text{MinPts} = 4$ as default of DBSCAN.

Discussion: Fig. 5 shows the effectiveness of our method on several different shapes. It works very well on especially cylindrical or elliptical shapes (Fig. 4) & (Fig. 5(a, c, g, h)). For the planar model in Fig. 5(b), our method generates very scattered skeletal candidates at the beginning as same as Medial Axis Transform does. However, through filtering and smoothing processes, low density regions are eliminated and most of pre-centered skeletal nodes are already at the center of the shape. Projecting and centering processes ensure that skeletal nodes locate at the center of the shape. Our result for this model is quite different from medial axis; but it shows another flavor for a curve skeleton on planar shapes.

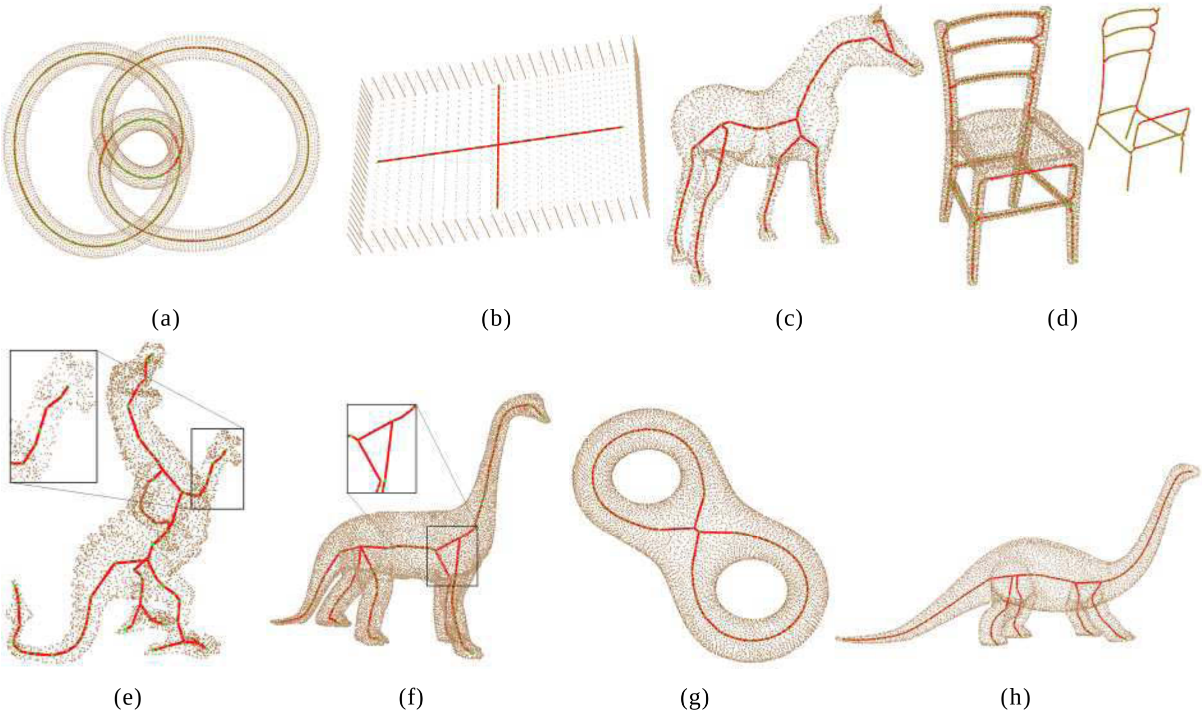


Fig. 5: Results of our method on various shapes including cylindrical and flat regions.

Our method is robust to noises (Fig. 6) because even though noisy points create noisy skeletal candidates or generate outliers, it will be well cleaned up through filtering and shrinking processes. Fig. 6(b), (d), (e) shows robustness of our method against noisy and incomplete data. We do not evaluate how much our method can resist to noises. The incomplete models Fig. 6(b) and Fig. 6(d) are created manually by removing some parts. The model Fig. 6(d) is missing almost half of the data. As for the noisy model Fig. 6(e), it is created by scattering approximately 30% of the whole points from its original boundary points. Even in such extreme condition (large missing data and high level of noises and outliers), our method extracts well the curve skeleton.

Time Complexity: The most time-consuming process in our algorithm is the computation of skeletal candidates. To speed up the computation, we utilize gridding technique (5 grids along the longest bounding box) to divide point cloud into smaller blocks and each computation uses several blocks through which the normal direction runs. Although the whole algorithm is implemented in C++, the gridding algorithm is not yet optimized. For 10K points, it takes approximately 6 seconds for skeletal candidates, 1.8 seconds for filtering and shrinking process, 2 seconds for thinning process, 0.9 seconds for centering process (ellipse fitting). Thus, the whole process takes approximately 7 seconds for 10K points, which are measured on a standard PC with Core i7-860 CPU.

Comparison: Fig. 6 shows the comparison of our algorithm with Laplacian-based contraction on point cloud [5] and Rotational Symmetry Axis (ROSA) [24]. Both methods extract good curve skeletons from point cloud model (Fig.6(c)). To the best of our parameter tuning for [5] and [24], the overall results show that our algorithm works better in term of centeredness and robustness to incomplete data and one with noises and outliers. It shows that [5] and [24] provide good curve skeletons on incomplete data (Fig. 6(b, d)) to some extent. [5] and [24] cannot generate a curve skeleton when some blocks of data are missing (Fig.6(b)) and fail to create a good curve skeletons with presence of very large missing data Fig.6(d). They both fails on noisy data with some outliers (Fig. 6(e)).

[5] extracts good curve skeleton from planar shape (Fig.6(a)), whereas [24] cannot define a curve skeleton on planar shapes as mentioned in the paper. On the other hand, even though our method generates, for planar shapes, a curve skeleton different from [5] and medial axis, it stays well under centerline definition.

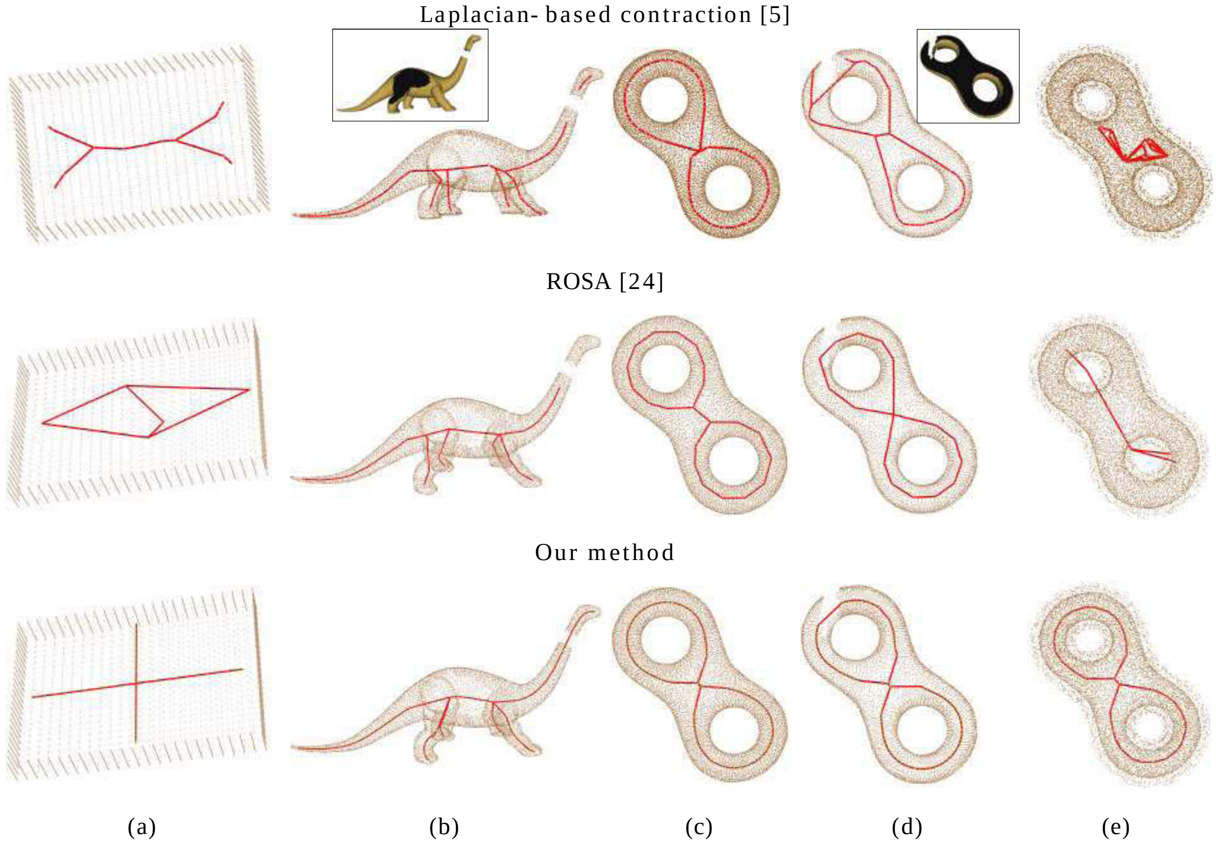


Fig. 6: Comparison of our method with Laplacian- based contraction [5] and Rotational Symmetry Axis (ROSA) [24].

Limitations: Our method provides very high accuracy on branch regions or cylindrical shapes Fig. 5(a, g). However, the accuracy tends to decline at join regions due to lack of skeletal nodes, for example at the front part of Dino model (Fig. 5(f)). This is due to the instability of cutting plane optimization, which results in incorrect neighborhoods for ellipse fitting. Consequently, skeletal nodes are largely eliminated at those joint regions.

Another limitation is that details on coarse parts of the model are sometimes reduced (Fig.5(e)) because our algorithm includes filtering, shrinking, and thinning processes. All of these processes tend to pull scattered skeletal candidates together to form highly concentrated nodes in which scattered points at the end part of the model are either eliminated or shrunk to some extent. Lowering local feature sizes $\theta_{shrink}, \theta_{mls}$ and density rate ρ may solve the problem.

Similar to the above limitation, skeletal candidates may be removed due to low density rates, for example, at thin planar shape of the model (Fig. 5(d)). Through filtering and shrinking processes, skeletal candidates at the central seating part of the model are mostly removed. However, cylindrical parts are well preserved and the frame of the seating part still exists because they are high density regions. The frame is, then, cleaned up through centering process because, far from theirs corresponding skeletal nodes, the centers of fitted ellipses appear at the center of seating part of the model. Consequently, centered skeletal nodes of the frame part are neglected.

5 CONCLUSIONS AND FUTURE WORK

We have proposed a direct extraction of curve skeleton from point cloud. Various models ranging from cylindrical to flat shapes of both clean and incomplete situation have been evaluated. The extracted curve skeletons are remarkable – robust to high noisy boundary points due to filtering and shrinking processes, and under centeredness-guaranteed property demonstrated by ellipse fitting. To achieve good results, parameter tuning is required; but it is an easy task done by modifying a local feature size and choosing the density rate within the size.

For future work, we would like to generate parameters automatically for our algorithm and implement it on raw scanning data with large missing parts or less scans per model. To exploit the computed curve skeleton for some applications such as surface reconstruction or shape matching is also our potential task.

ACKNOWLEDGEMENTS

The availability of source code [5] and [24] is of much help in the comparison section. Models used in this paper are obtained from the AIM@SHAPE repository.

REFERENCES

[1] Amanta, N.; Bern, M.; Eppstein, D.: Crust and the Beta-Skeleton: Combinatorial Curve, Graphical Models and Image Processing, 60(2), 1998, 125-135. DOI:10.1006/gmip.1998.0465

[2] Au, K.-C. O.; Thai, C.-L.; Chu, H.-K.; Cohen-Or, D.; Lee, T.-Y.: Skeleton Extraction by Mesh Contraction, ACM Transactions on Graphics, 27 (3), 2007, 44:1-44:10. DOI:0.1145/1399504.1360643

[3] Bertrand, G.; Aktouf, Z.: Three-Dimensional Thinning Algorithm Using Subfields. Vision Geometry III, Proc. 2356, 1994, 113-124. DOI:10.1109/TPAMI.2002.1114851

[4] Blum, H.: A Transformation for Extracting New Descriptors of Shape, In Models for the Perception of Speech and Visual Form, MIT Press, 1967, 362-380.

[5] Cao, J.; Tagliasacchi, A.; Olson, M.; Zhang, H.; Su, Z.: Point Cloud Skeletons via Laplacian-Based Contraction, Shape Modeling International Conference (SMI), 2010, 187-197. DOI: 10.1109/SMI.2010.25

[6] Cornea, N. D.; Demirci, M. F.; Silver, D.; Shokoufandeh, A.; Dickinson, S. J.; Kantor, P. B.: 3D Object Retrieval Using Many-to-many Matching of Curve Skeletons, Shape Modeling and Applications 2005 International Conference, 2005, 366-371. DOI:10.1109/SMI.2005.1

[7] Cornea, N. D.; Silver, D.; Min, P.: Curve-Skeleton Properties, Applications and Algorithms, IEEE Transactions on Visualization and Computer Graphics, 13 (3), 2007, 530-548. DOI:10.1109/TVCG.2007.1002

[8] Dey, T. K.; Sun, J.: Defining and Computing Curve Skeletons with Medial Geodesic Functions, SGP '06 Proceedings of the Fourth Eurographics Symposium on Geometry Processing, 2006.

[9] Everitt, B.; Landau, S.; Leese, M.; Stahl, D.: Cluster Analysis, In Miscellaneous Clustering Methods, A John Wiley and Sons, 5 ed., 2011.

[10] Fitzgibbon, A.; Pilu, M.; Fisher, R. B.: Direct Least Square Fitting of Ellipses, IEEE Transactions on Pattern Analysis and Machine Intelligence, 21 (5), 1999, 475-480. DOI:10.1109/34.765658

[11] Halif, R.; Flusser, J.: Numerically Stable Direct Least Squares Fitting of Ellipses, The Sixth International Conference in Central Europe on Computer Graphics and Visualization, 21 (5), 1998, 125-132.

[12] Hassouna, M. S.; Farag, A. A.: Robust Centerline Extraction Framework Using Level Sets, In IEEE Conference on Computer Vision and Pattern Recoginition (CVPR), 2005, 458-465. DOI:10.1109/CVPR.2005.306

[13] Hoppe, H.; DeRose, T.; Duchamp, T.; McDonald, J.; Stuetzle, W.: Surface Reconstruction from Unorganized Points, Proc. ACM SIGGRAPH, 1992, 71-78. DOI:10.1145/142920.134011

[14] Huang, H.; Li, D.; Zhang, H.; Ascher, U.; Cohen-Or, D.: Consolidation of Unorganized Point Clouds for Surafce Reconstruction, ACM Transactions on Graphics, 28 (5), 2009. DOI:10.1145/1618452.1618522

[15] Lee, I.-K.: Curve Reconstruction from Unorganized Points, Computer Aided Geometric Design, 17 (2), 2000, 161-177. DOI:10.1016/S0167-8396(99)00044-8

[16] Levin, D.: The Approximation Power of Moving Least-Squares, Mathematics of Computation, 67 (224), 1998, 1517-1531.

[17] Ma, C.-M.; Wan, S.-Y.; Lee, J.-D.: Three-Dimensional Topology Preserving Reduction on the 4-Subfields, IEEE Trans. on Pattern Analysis and Machine Intelligence, 24 (12), 2002, 1594-1605. DOI:10.1109/TPAMI.2002.1114851

[18] Ogniewicz, R.; Ilg, M.: A Multiscale MAT from Voronoi Diagrams: the Skeleton-Space and its Application to Shape Description and Decomposition, Aspects of Visual Form Processing, World Scientific, 1994.

[19] Ogniewicz, R.; Kubler, O.: Voronoi Skeletons: Theory and Applications, IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 1992, 63-69. DOI:10.1109/CVPR.1992.223226

[20] Shapira, L.; Shamir, A.; Cohen-Or, D.: Consistent Mesh Partitioning and Skeletonisation Using the Shape Diameter Function, Visual Computing, 24 (4), 2008, 249-259. DOI:10.1007/s00371-007-0197-5

[21] Sharf, A.; Lewiner, T.; Shamir, A.; Kobbelt, L.: On-The-Fly Curve-Skeleton Compuation for 3D Shapes, Computer Graphics Forum, 26 (3), 2007, 323-328.

[22] Siddiqi, K.; Pizer, S. M.: Medial Representations : Mathematics, Algorithms and Applications, Springer, 2009.

[23] Svensson, S.; Nystrom, I.; Sanniti di Baja, G.: Curve Skeletonization of Surface-Like Objects in 3D Images Guided by Voxel Classification, Pattern Recognition Letters, 23, 2002, 1419-1426.

[24] Tagliasacchi, A.; Zhang, H.; Cohen-Or, D.: Curve Skeleton Extraction from Incomplete Point Cloud, ACM Transactions on Graphics, 28 (3), 2009, 71:1-71:9. DOI:10.1145/1531326.1531377

[25] Vollmer, J.; Mencl, R.; Muller, H.: Improved Laplacian Smoothing of Noisy Surface Meshes, Computer Graphics Forum, 18 (3), 1999, 131-138.