A Centered Curve Skeleton Extraction from 3D Point Cloud

The University of Tokyo 〇 Vanna Sam, Hiroaki Kawata, Takashi Kanai

A curve skeleton of 3D complex shape is a compact and simple graph, which provides sufficient information on geometry and topology of the shape. It is very useful for many computer graphics applications involving shape analysis. Much research has been focused on volumetric and polygonal mesh models. However, only a few have been paid attention to curve skeleton extraction from point clouds; particularly from noisy and incomplete data. We propose a robust algorithm for extracting curve skeleton from point cloud. The process starts from estimating centers of antipodes of each point, so called skeletal candidates. Those candidates are absolutely inside the shape; but scattered. We thus filter and shrink them to create less noisy skeletal candidates before applying one-dimensional Moving Least Squares to build a line-like point cloud. We then downsample the computed thin cloud to sparse skeletal nodes. We finally utilize least squares ellipse fitting to relocate skeletal nodes to guarantee that they are under centeredness property of a curve skeleton. We demonstrate the consistency of our method on several models ranging from clean to noisy and incomplete data.

Keywords: Curve Skeleton, point cloud, antipode, regression line, ellipse fitting

1. INTRODUCTION

A curve skeleton, 1D curve is one of the most important structures of 3D complex shape, which provides sufficient information on geometry and topology of the shape. Even though, there is no absolute definition for a curve skeleton [3], the most commonly targeted curve skeleton is a one-dimensional polysegment 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 graphs applications which involve in shape analysis such as shape recognition, object matching and retrieval, animation transfer, and so forth. These applications correspondingly require several necessary properties among topology preservation, centeredness, robustness, connectedness, reliability, hierarchy and reconstructability [3].

The major problem in extracting a curve skeleton is how to interpret 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, which require full knowledge of models. Some have been also paid attention to both polygonal mesh and point cloud models. However, only a few have been focused on noisy and incomplete data models. Recently, [2] utilizes Laplacian technique to contract boundary points to create a curve skeleton. Each point requires a local creation of one-ring connectivity of neighborhood for Laplacian operation and much attention also has to be paid to contraction parameters. This approach 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. [7] applies cutting plane technique 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.

We propose a robust computation for curve skeleton from point clouds including 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 process. 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 and guarantee that they lie on the center of the model i.e. our curve skeleton is under robustness and centeredness properties.

We main contributions include (i) a direct extraction of curve skeleton from point cloud, (ii) a robust extraction from noisy and incomplete data and (iii) a centeredness-guarantee curve skeleton defined by ellipse fitting.

2. CURVE SKELETON EXTRACTION

Our algorithm proceeds as follows. Given an oriented point cloud, we first attempt to compute the center of antipodes of each point in the cloud. The computed centers are called skeletal candidates. They are basically scattered and theoretically laid 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. We thus remove any skeletal candidate of low density. The remaining skeletal candidates pass through Laplacian smoothing to form a strong-bond distribution. We then process 1D Moving Least Square [6] 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.

2.1 Antipodes and Skeletal Candidates

Two points are antipodes if they are diametrically opposite. Let \( p_i, n_i \in P \) be a point and its normal vector in the point cloud \( P \). Different from [6], we send only one ray from \( p_i \) in reverse-normal direction until we find a point on the opposite side of the boundary called an antipodal point \( q_i \) (See Fig.1-a). In the perfect case, the normal of \( q_i \) and \( n_i \) lie on the same direction; but in real case, it is hard to find. We thus adopt \( q_i \) as an antipode if and only if its respects three conditions. (1) is that its normal forms an obtuse angle with \( n_i \), (2) \( q_i \) lies close to the direction of \( n_i \), and (3) \( q_i \) provides the smallest distance with \( p_i \). The center of the antipode, skeletal candidate \( SC_i \), is located at a half of the smallest distance and lies on the normal direction \( n_i \). Skeletal candidate \( SC_i \) is under two constraints:

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

2.2 Filtering and Thinning Process

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. This can be obtained by projecting \( SC_i \) onto their regression lines using Moving Least Squares (MLS) technique [6]. Beforehand we need a clean distribution of skeletal candidates. We thus eliminate any \( SC_i \) of low
density neighborhood \( p < \rho \) and shrink them to create less noisy skeletal candidates with Laplacian smoothing within a local feature size \( \theta_{\text{shrink}} \) (Fig.1-d). A prescribed density rate \( \rho \) and the feature size \( \theta_{\text{shrink}} \) 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. In our case, we compute 3D regression line by using the Principle Component Analysis (PCA). The regression line passes through the centroid of PCA and directs along the eigenvector of the largest eigenvalue of the covariance matrix of PCA. A local support size for PCA is \( \theta_{\text{mean}} \). We iteratively project each point \( X_i \) onto its 3D regression line until 1D point cloud is obtained (Fig.2-a). The process is repeated four times through our experiment and results are sufficient enough for skeletal nodes sampling. The sampling is done by picking the farthest point in the sample as a skeletal node and by eliminating its remaining neighbor points from 1D cloud. The process is continued until no skeletal candidate is left in the stack (Fig.2-b).

The incomplete models (Fig.3-d,g) are manually created. The model (Fig.3-g) is missing almost half of the data. Even in such extreme condition, our method extracts well the curve skeleton.

**2.3 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 its centeredness. We thus utilize least squares ellipse fitting in [4] to relocate skeletal nodes. To obtain nodes in 2D space for ellipse fitting, we utilize cutting plane idea introduced in [7] to collect nodes close to plane within a thickness \( \theta_{\text{shrink}} \) and to extract relevant neighborhood, we apply Density-Based Spatial Clustering of Application with Noise (DBSCAN). We refer the details of ellipse fitting problem to [4]. The center of each fitted ellipse will be considered as correct node if it locates close to its corresponding skeletal node with a distance \( e_{\text{center}} \). Otherwise, it is neglected (Fig.2-c).

**Curve Extraction:** To construct a curve skeleton, we apply nearest neighbor crust algorithm [1] on centered skeletal nodes (Fig.2-d).

**3. RESULTS AND DISCUSSION**

We have demonstrated our algorithm on different models including clean, noisy and incomplete data. For efficiency reason, we downsample models to approximately 10K points.

**Parameters:** Parameters 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 – \( \rho \) and \( \theta_{\text{shrink}} \). In this paper, we choose \( \rho \) from 6 to 18 points (a point whose neighboring points are less than \( \rho \) is considered as outlier) and \( \theta_{\text{shrink}} \) is set from 1.5 to 2.5% of the diagonal of the bounding box. Other parameters are estimated as follows: \( \theta_{\text{sample}} = \theta_{\text{plane}} = e_{\text{center}} = \theta_{\text{shrink}} \theta_{\text{mean}} = 2\theta_{\text{shrink}} \), and parameters for DBSCAN are \( eps = \theta_{\text{shrink}} \) and \( MinPts = 4 \).

**Results:** Fig.3 shows the effectiveness of our method on several different shapes. It works very well on especially cylindrical or elliptical shapes (Fig.3-a,c,e,f). For the planar model in Fig.3-b, our method generates a curve skeleton different from medial axis. However, it shows another flavor for a curve skeleton on planar shape.

Our method is robust to noises (Fig.3-h) because even though noisy points create noisy skeletal candidates or generate outliers, it will be well cleaned up through filtering and shrinking processes. The model (Fig.3-h) is created by scattering approximately 30% of the whole points from its original boundary points.

**Time Complexity:** The most time-consuming process in our algorithm is the computation of skeletal candidates. To speed up the computation, we utilize spatial partitioning 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 spatial partitioning algorithm is not yet optimized. For 10K points, it takes approximately 5 seconds for skeletal candidates, 0.2 seconds for filtering and shrinking process, 2.3 seconds for thinning process, 0.2 seconds for node sampling, and 0.9 seconds for centering process (ellipse fitting). Thus, the whole process takes approximately less than 9 seconds for 10K points.

**Limitations:** Our method provides very high accuracy on branch regions or cylindrical shapes. However, the accuracy tends to decline at joint regions due to lack of skeletal nodes, for example at the front part of Dino model (Fig.3-e). The cause of this problem is due to the instability of cutting plane optimization, which results incorrect neighborhoods for ellipse fitting. Consequently, skeletal nodes are largely eliminated at those joint regions.

**4. CONCLUSIONS AND FUTURE WORK**

We have proposed a direct extraction of curve skeleton from point cloud data. The extracted curve skeletons are remarkable - robust to high noises and under centeredness-guarantee property.

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.

**REFERENCE**


