Skeleton-Growing:A Vector-Field-Based 3D Curve-Skeleton Extraction Algorithm

Natapon Pantuwong*
University of Tokyo

Masanori Sugimoto†
University of Tokyo

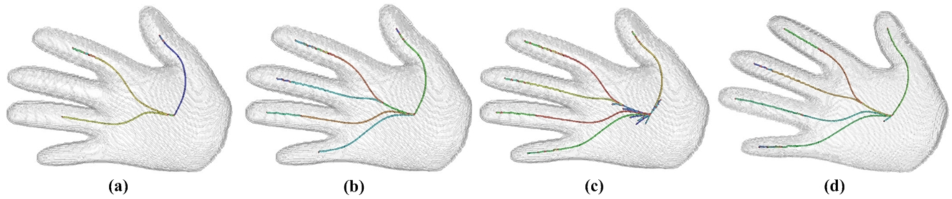

Figure 1: *Curve-skeleton extracted by connecting critical points and low-divergence points with the divergence threshold of 0%(a), 20%(b), 25%(c). Curve-skeleton extracted by the proposed method(d).*

Abstract

Vector-field based method is one of the 3D curve-skeleton extraction algorithms. Typically, critical points in the vector field inside 3D object are connected to form the curve-skeleton. However, critical points usually does not distribute for all important part of the 3D object. Therefore, other features are used to produce a reliable result. Although this strategy can deliver a curve-skeleton which captures all important part, the curve-skeleton usually comes with unnecessary segments. This paper proposes the skeleton-growing algorithm which automatically produces the curve-skeleton with small amount of such segments. It searches for a set of high-curvature boundary voxels as starting points to find a set of suitable seed points which will be used to grow the curve-skeleton. We propose an unnecessary segment removal algorithm which can reduce the skeleton-noise density. A direction-selection algorithm is developed to avoid searching in irrelevant directions. The proposed method can produce single reliable result curve-skeleton which could be applied in many different applications including matching, animation, and visualization.

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

Keywords: skeleton-growing, curve-skeleton, vector-field, curvature

1 Introduction

A curve-skeleton is a thinned-line representation which captures the essential topology of a 3D object. It is useful in many applications, such as 3D object recognition, visualization and computer animation. There are many skeletonization algorithms in the literature. A very good review of curve-skeleton extraction methods can be found in [Cornea et al. 2007].

According to [Cornea et al. 2007], vector-field-based method could deliver a very good result. A curve-skeleton could be extracted by connecting all critical points (the points where the magnitude of the vector vanishes) together. However, the curve-skeleton produced by using this strategy usually contains only some segments of the expected skeleton as shown in Figure 1(a). Therefore, the missing part must be added by considering other important features of vector field. For example, the method in [N.D.Cornea et al. 2005] connects a set of low-divergence points to complete the skeleton. Although this method could provide a correct curve-skeleton, it is difficult to identify the optimal threshold which provides the expected result. For example, in Figure 1(b), a curve-skeleton, which is extracted by using divergence threshold of 20%, has no skeleton-noise, but there are some missing segments (near the tip of some fingers). In Figure 1(c), the divergence threshold is increased to 25% which results in a curve-skeleton that contains all important part including skeleton-noise. This example shows the threshold-adjusting problem of the previous work.

The proposed skeleton-growing algorithm is developed to come over this problem. This method could produce a single reliable result, as shown in Figure 1(d). In contrast to previous work[N.D.Cornea et al. 2005], this method searches for a set of high curvature boundary voxels, and use them to search for a set of seed points. After that, this method uses seed points to grow a result curve-skeleton by tracing the vector field in the direction of the eigenvectors of Jacobian matrix of the vector field at each seed point. Some of tracing paths are determined as skeleton segment by the proposed unnecessary-segment removal algorithm. By using such algorithm, the extracted curve-skeleton captures all essential part with low density of skeleton-noise. Apart from vector-field creating step, the proposed method does not require to calculate at every voxel which makes the processing time of the proposed method is comparable to the previous work.

2 The Skeleton-Growing Algorithm

The proposed skeleton-growing algorithm is described as follows:

1. Use a method of choice to calculate the vector field inside the 3D object. In this paper, we use the repulsive force field which is explained in [N.D.Cornea et al. 2005].
2. Calculate the mean-curvature value[Kindlmann et al. 2003] for every boundary voxel. The high-curvature threshold is the mean value of high peak of the curvature histogram. Example of high-curvature points is shown by the red area of Figure 2(a).
3. Sort all high-curvature voxels in descending order by curvature value.For each high-curvature voxel, trace the vector field in the direction of the vector at that point until reach the critical point. We calculate the divergence value for every voxel during this tracing process. Every critical point will be used as a seed point to grow the skeleton. Furthermore, some of tracing paths are determined as skeleton segments by using the condition described below.

*e-mail: na@itl.t.u-tokyo.ac.jp
†e-mail: sugi@itl.t.u-tokyo.ac.jp

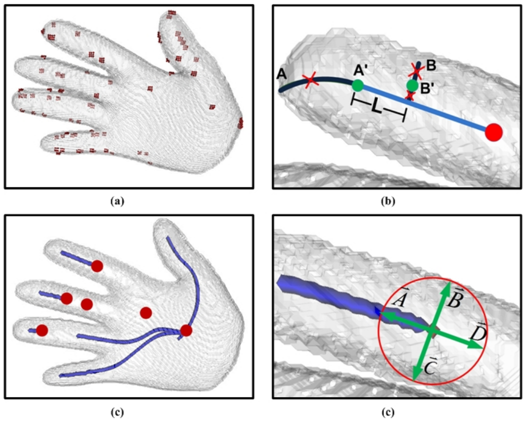

Figure 2: *Skeleton-growing algorithm. (a):Set of high-curvature points, (b):Unnecessary-segment removal algorithm, (c):Critical points and skeleton segments extracted by using high-curvature points, (d):Direction-selection algorithm*

- For every tracing path that does not pass any previous visited voxels, determine the divergence value of all traced voxels from starting point. If the divergence value continuously decreases and reaches the steady state, the position of steady state should be a point that the vectors tend to converge, and the tracing path from that point to critical point is determined as a skeleton segment. Otherwise, the tracing path is ignored. In Figure 2(b), if we trace the vector field from point *A*, the divergence value will reach steady state at point *A´*. Therefore, the segment from *A´* to critical point (red point) is determined as skeleton segment, and the segment from *A* to *A´* is removed.
- For every tracing path that meets any previous visited voxels, the tracing process is stopped at that meeting point. Then use the same condition to determine that tracing path. Therefore, in Figure 2(b), the segment from *B* to *B´* is ignored. However, we also determine the distant of every segment that passes the condition. If the distance of that segment is longer than half of the distance from the meeting point to its parents node(*L* in Figure 2(b)), it will be determined as a skeleton segment. Therefore, the segment from *B´* to the meeting point is also removed.

Figure 2(c) shows critical points(red) and skeleton segments(blue) which are extracted using the condition explained above.

4. Classify each critical point to saddle or attracting point. For saddle point, trace the vector filed in the direction of eigenvectors of Jacobian matrix of vector field which corresponding to the positive eigenvalues and their reverse directions. For attracting point, all eigenvectors are pointing to that point. Therefore, we have to trace the vector field in all direction of eigenvectors and their reverse directions, which may include some irrelevant directions. To avoid this problem, we create a sphere with a radius equal to the distance to the nearest boundary. If the position of the crossing point between the direction vector and the surface of sphere is near enough to the boundary, this direction ($\vec{B}$ and $\vec{C}$ in Figure 2(d)) is ignored. If the direction vector is pointing to the direction that has already been processed ($\vec{A}$ in Figure 2(d)), this direction is also ignored. For every searching path that meets a new critical point or previously created skeleton, that path will be determined as a skeleton segment.
5. Repeat step 4 for every newly found critical point until there are no more new critical points.

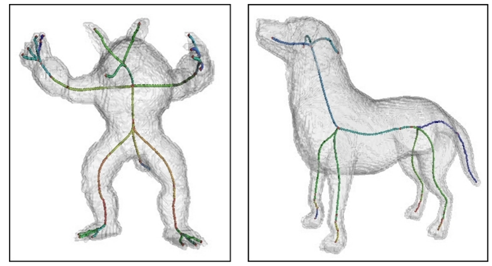

Figure 3: *Examples of curve-skeletons extracted by the proposed method*

3 Experimental Result

We extracted the skeleton from 30 different voxelized 3D models with the resolution around 80^3 voxels. Figure 3 shows an example of a resulting curve-skeleton extracted by the proposed method. The computation time of the proposed method is, on average, 10.087 seconds which is comparable to the previous work[N.D.Cornea et al. 2005] that takes 8.981 seconds. We measure the recall (the ratio of the number of retrieved correct skeleton segment to the total number of correct skeleton segment) and the precision (the ratio of the number of retrieved correct skeleton segment to the total number of retrieved skeleton segment) from the result of both previous work and the proposed method. For the previous work, with the divergence threshold of 20%, the recall is 88.66%, and precision is 100%. Although it does not produce any skeleton-noise, there are some missing segments. For the divergence threshold of 25%, the recall is 92.85%, and the precision is 78.61%. The recall value is improved, but the result curve-skeleton comes with skeleton-noise. For the proposed method, it produces only single result with recall of 100% and precision of 94.38%. The result is correct with less density of skeleton-noise compared to the result of previous work.

4 Conclusion

This paper introduces the skeleton-growing algorithm that can be used to extract the curve-skeleton from the 3D object automatically. In contrast to previous work, this method does not require the threshold adjusting process, while still achieving good accuracy. The processing time is comparable to the previous work because the proposed method does not search for skeleton segment in every voxel of the 3D object. The resulting curve-skeleton comes with very small density of noise which makes it is suitable to use with any applications. For our future work, we are planning to develop an automatic animation system by using the result curve-skeleton.

References

CORNEA, N., ET AL. 2007. Curve-skeleton properties, applications, and algorithms. *Visualization and Computer Graphics*.

KINDLMANN, ET AL. 2003. Curvature-based transfer functions for direct volume rendering: Methods and applications. In *VIS*.

N.D.CORNEA, ET AL. 2005. Computing hierachical curve-skeletons of 3d objects. *The Visual Computer 21*, 11, 945–955.