3D Curve-Skeleton Extraction Using a Skeleton-Growing Algorithm

Natapon Pantuwong* Masanori Sugimoto†

University of Tokyo

ABSTRACT

A curve skeleton is a line-like representation of 3D objects. This paper describes a skeleton-representation algorithm that extracts the skeleton from the vector field inside the 3D object. In contrast to previous work, the vector field is created by using a pseudo-normal vector, which requires less computation time than a typical repulsive force function. The skeleton can be extracted by analyzing the vector-field topology. The proposed skeleton-growing method does not require calculations for every voxel in the critical-point-determining process. It searches for one of the critical points in the vector field and uses the direction of vector at that point to search for other critical points. We also propose a direction-selection algorithm to avoid searching in irrelevant directions. The proposed method requires significantly less computation time than previous vector-field-based approaches while still achieving good accuracy.

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

1 INTRODUCTION

A curve skeleton is a thinned-line representation of a 3D object, which is useful in many visualization tasks. There are many skeleton-extraction methods in the literature, which can be categorized as being either a volumetric approach or a geometric approach. A volumetric approach [3, 4] calculates the information inside the object, making it suitable for 3D volumetric objects. A geometric approach [6] extracts a curve skeleton from a 3D mesh object. However, a volumetric object can be converted into a mesh object, and vice versa, so we can apply either approach to either type of 3D object.

The proposed algorithm is developed using a vector-field-based method (volumetric approach). In contrast to previous work [3, 4], our method creates a vector field from the pseudo-normal vector, which requires less computation time than a typical repulsive force function, while still providing a correct skeleton. We propose a skeleton-growing algorithm that does not search for critical points at every voxel. It detects one of the critical points in the vector field and uses it as a seed point to search for other critical points. We also propose a direction-selection method to avoid searching in irrelevant directions. This can significantly reduce the computation time compared with other vector-field-based algorithms. The proposed method can be applied to many applications that require rapid processing in the skeletonization step.

2 PSEUDO-NORMAL VECTOR FIELD

Typically, the vector field inside a 3D object is created by a repulsive force function[3, 4]. Although this method provides a good shape for the curve skeleton, it requires much computation time because it uses all boundary voxels in each calculation. From our observations, this can be estimated by creating a normal vector, which points into the object, at each voxel. Figure 1(a) shows how the algorithm works. At each iteration, we create a normal vector for each boundary voxel (marked *B*). Then, the interior neighboring points (marked *I*) will become the new boundary points in the next iteration. This calculation is performed iteratively until there is no boundary voxel that has an interior neighboring point. To simplify the calculation, we use the pseudo-normal vector method[5] to estimate the normal vector. The pseudo-normal vector is approximately perpendicular to the tangent plane at the given point and points inside the object. Figure 1(b) shows how to generate a pseudo-normal vector. Unit vectors (red vectors) are created for every direction that points to interior neighboring voxel(green). A pseudo-normal vector (blue vector) is then obtained by summing and normalizing them. An example of a pseudo-normal vector field is shown in Figure 1(c).

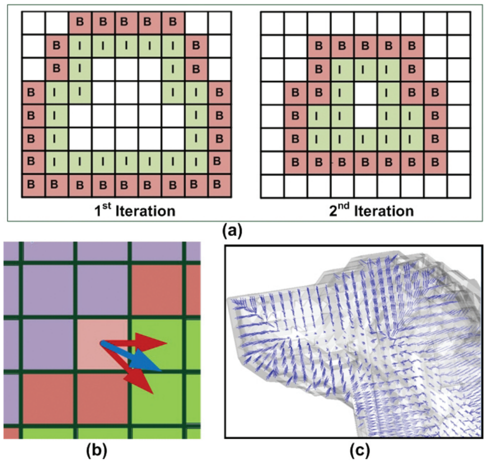

Figure 1: (a)The boundary points move inward during the vector-field calculation. (b)Pseudo-normal vector calculation at boundary voxel. (c)Pseudo-normal vector field.

3 THE SKELETON-GROWING ALGORITHM

A curve skeleton can be extracted from the vector field by connecting all the critical points in the vector field. Typically, we need to determine the critical point condition for every voxel. We develop a skeleton-growing algorithm to reduce the number of such calculations, which is described as follows:

1. Calculate the divergence value[1] for each voxel and then search for the voxel that has the minimum divergence value. If it is a critical point, use it as a seed point. Otherwise, trace the vector field in the direction of the vector at that point until a critical point is found, and use it as a seed point.

2. Calculate the eigenvectors at the critical point[2]. Then trace the vector field in the directions of the eigenvectors of Jacobian matrix of the vector field at that critical point and in their reverse directions. We save the critical-point-determining results for every voxel along the search path to avoid redun-

---

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

†e-mail: sugi@itl.t.u-tokyo.ac.jp

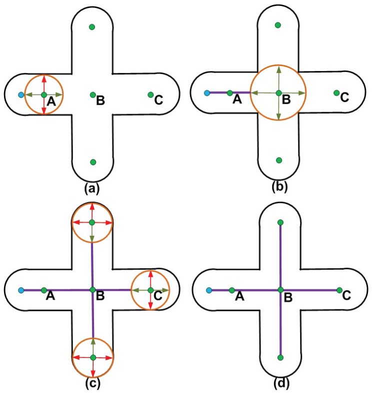

Figure 2: Skeleton growing with the direction-selection algorithm.

dant calculations. For any search path that meets a new critical point, this path will be a skeleton segment. If a search path reaches the boundary, this path will be ignored. However, we determine the divergence value for every voxel during the search process. If any voxels have a divergence lower than a threshold value, these voxels will be set as a skeleton point, and this segment will be determined as a skeleton segment. In our development, the threshold value was set to $Div_{min} + (Div_{dif} \times percentage)$, where Div_{min} is the minimum divergence value, Div_{dif} is the difference between the maximum and minimum divergence values, and *percentage*,which is user-defined, is used to control the *threshold*.

3. Repeat step 2 for every newly found critical point until there are no more new critical points.

4. Connect the remaining voxels that have a divergence value lower than the threshold to the constructed skeleton.

We also propose a direction-selection algorithm to reduce the number of search paths at each critical point. At each critical point, 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 can be ignored. The distance to the nearest boundary can be estimated from the number of iterations used during the creation of the pseudo-normal vector field. That is, in Figure 1 (a), the distance to the nearest-boundary voxels for all voxels marked as B is 0 in the first iteration, the distance is 1 in the second iteration, and so on.

Skeleton growing is demonstrated by Figure 2. The critical point A is the seed point. There are four possible search directions. Two directions are determined as irrelevant by the condition described above (red vectors), with the other two directions being relevant directions (green vectors). Searching to the left, we will reach the boundary. However, during this tracing process, we find a low-divergence point (blue point), so the segment between this point and A is determined as a skeleton segment (purple line). Searching to the right, we will reach the critical point B. At this point, the method does not eliminate any directions, because none are irrelevant (Figure 2(b). However, we can ignore the left direction because it has already been searched. Therefore, we process each of the other three search paths as before. We obtain the curve skeleton shown in Figure 2(d). Although there remain some irrelevant directions that are not eliminated, such as the right direction at critical point C (Figure 2(c)), this method does remove almost all of the irrelevant directions while retaining all relevant directions.

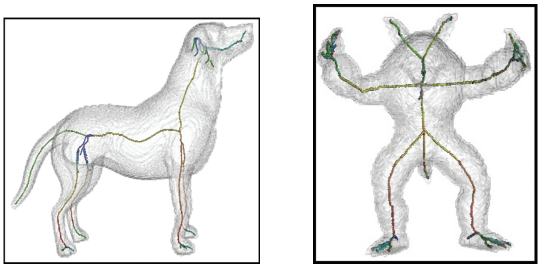

Figure 3: Examples of curve skeletons extracted by the proposed method with a *percentage* value of 25%.

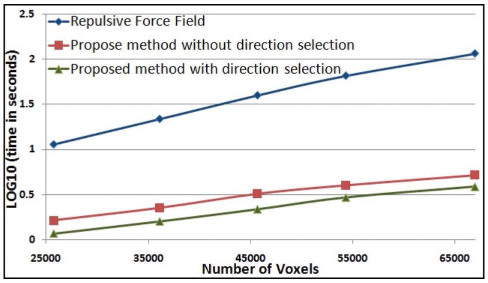

Figure 4: Comparison of computation times (log scale) for the proposed method and the previous method.

4 EXPERIMENTAL RESULTS

We extracted the skeleton from 30 different voxelized 3D models. The resolution of each model is around 80^3 voxels. Figure 3 shows an example of a resulting skeleton for a *percentage* value, which is used to control the *threshold* as mentioned in previous section, of 25%. We found that a *percentage* value of around 20-25% is sufficient to produce reliable result. Our direction-selection method can remove, on average, 92% of the irrelevant directions while retaining the relevant directions. The proportion of relevant directions among selected directions is 84.98%, while the proportion for a skeleton-growing method without direction selection is 30.28%. Figure 4 compares computation times for our proposed method and the repulsive-force-field method.

5 CONCLUSION

This paper introduces a skeleton-growing algorithm that can be used to extract a curve skeleton from a 3D object. The proposed method can reduce the computation time because it does not search for critical points at every voxel. We also propose a search-direction selection algorithm to avoid irrelevant search paths. The experimental results confirm that our proposed method can be used to extract a correct skeleton while significantly reducing the computation time required by previous methods. In our future work, we are planning to use local surface information to eliminate all irrelevant search directions.

REFERENCES

[1] Bouix et al. Divergence-based medial surfaces. In *ECCV '00*, pages 603–618, London, UK, 2000. Springer-Verlag.

[2] A. Globus et al. A tool for visualizing the topology of three-dimensional vector fields. In *Proc. IEEE Vis'91*, pages 33 –40, 1991.

[3] P.-C. Liu et al. Automatic animation skeleton using repulsive force field. In *Proc. 11th Pacific Conf.*, pages 409 – 413, 8-10 2003.

[4] N.D.Cornea et al. Computing hierachical curve-skeletons of 3d objects. *The Visual Computer*, 21(11):945–955, 2005.

[5] N. Pantuwong and M. Sugimoto. 3d curve-skeleton extraction algorithm by using psuedo-normal vector field. submitted for publication.

[6] M. Poirier and E. Paquette. Rig retargeting for 3d animation. In *GI '09: Proceedings of Graphics Interface 2009*, pages 103–110, 2009.