Journal of Computational Information Systems 9: 11 (2013) 4469–4478
Available at http://www.Jofcis.com

Extracting Curve-Skeleton for Use in Shape Deformation

Long YANG [1], Zhiyi ZHANG [1,*], Dongjian HE [2], Yuan DAI [1]

[1] *College of Information Engineering, Northwest A&F University, Shaanxi 712100, China*

[2] *College of Mechanical and Electronic Engineering, Northwest A&F University, Shaanxi 712100, China*

Abstract

In order to generate shape deformation, in this paper, we present a geometric method to extract curve-skeleton from point clouds of cylinder similar shapes. Our method is implemented through searching the optimal cutting plane for each point on the surface of an object. It determines the optimal cutting plane only through two sets of rotations from the initial cutting plane. Experiments show that our skeleton extracting method is effective for cylindrical objects. In addition to generating curve-skeleton, by changing the skeleton poses we produce different deformed results of the original shapes as well.

Keywords: Curve-Skeleton; Point Clouds; Cylindrical Shapes; Optimal Cutting Plane; Shape Deformation

1 Introduction

Skeleton as an effective descriptor of shape is probably first proposed in [1]. It used 1D curve as a compact form to simply represent the 2D or 3D shapes. Curve-skeleton explicitly carries essential topological and geometric information of an object. We can intuitively see the structural character of an object through its 1D curve-skeleton. Therefore, Skeleton has been widely used in many 3D application domains, just like shape representation [2, 3, 4], shape recognition and matching [5, 6], shape deformation [7, 8], computer animation [9, 10] and many other fields.

Skeleton is extracted primarily either from solid models or continuous polygonal meshes. There are many approaches used in shape skeletonization. A common classification has assorted these existing approaches to several classes according to their different implemental principles [11, 12]. They are thinning (or boundary propagation) techniques [13], distance-based approaches [14] and geometric methods [15, 16].

The preceding two classes of methods mainly process solid models in discrete voxel space. While geometric methods mostly extract skeleton from polygonal meshes [6, 17, 18, 19]. All these methods have received specific effects with respect to different applications. Besides, there are also some other geometric methods that produce skeleton directly from point clouds models

*Corresponding author.
Email address: `liurenzu@sohu.com` (Zhiyi ZHANG).

1553–9105 / Copyright © 2013 Binary Information Press
DOI: 10.12733/jcis6169
June 1, 2013

[16, 20, 21]. However, the methods extracting skeleton from point clouds data are much less than those from the solid and polygonal models. Consequently it is necessary to explore more efficient skeleton extracting methods from scattered point clouds data.

In many application fields, cylindrical shape occupies an important position for its relative simplicity. Many objects and their branches can be approximately taken as the generalized cylinders. Therefore, how to produce curve-skeleton from a cylindrical shape becomes a primary topic in practical use.

In this paper, we use an intuitive and simple geometric method to extract curve-skeleton directly from point clouds data of the cylindrical shape. Next section we will discuss the methods of skeleton extraction from point clouds data. In section 3, our skeleton extracting method, the Optimal Cutting Plane approach, will be introduced. Examples of both curve-skeleton extraction and skeleton driven deformation are demonstrated in section 4 and section 5 respectively. In final, we present conclusions and the future work.

2 Related Work

Many laser scanners can capture high quality point clouds for 3D models. Extracting curve-skeleton from these discrete points provides an effective way for further shape manipulations.

The level sets algorithm was used to skeletonize the polygonal meshes in [4, 22]. While the algorithm based on level sets also works properly in point clouds models [21]. For the complexity of the algorithm which processes scattered points data directly, it has to involve some preprocessing to generate the contiguous lever sets.

This algorithm specifies one point as the source of a shape and calculates the shortest path distance from each surface point to the source point. And the shortest path distance was used to create the neighborhood graph for constructing the level sets. Although this distance keeps connectivity of adjacent skeleton points corresponding to contiguous level sets, the processing is costly.

In addition, for some cases, it generates identical orientation level sets for branches with different directions so that this gives rise to improper correspondence between skeleton point and shape surface points. The improper correspondence inevitably obstructs the further use in shape deformations.

In [23], the authors produced curve-skeleton from point clouds of the 3D objects. It calculates skeleton point based on the rotational symmetry axis of an oriented point set. Furthermore this algorithm has been used to retrieve the missing data from incomplete cylindrical shapes and received nice performance. However obtaining an oriented point set lying on an optimal cutting plane is computational complex.

In this paper, we limit our discussion only to how to extract curve-skeleton from scattered points of cylindrical objects. Our method uses the optimal cutting plane which passes through a given surface point to determine its corresponding skeleton point. The processing of searching for the optimal cutting plane, which is more intuitive and simpler than the method used in [23], and the processing of skeletonization will be presented in detail in the subsequent section.

3 Skeletonization

There are numerous planes passing through a specified point in three-dimensional space. Once a surface point is given, our method can determine its optimal cutting plane by implementing two sets of rotations sequentially.

The optimal cutting plane of a surface point can be defined as follows: all planes (each has a small constant thickness λ and passes through the specified point) cut a cylindrical object respectively. The plane with the minimal sectional slice area is selected as the optimal cutting plane for the specified surface point.

Certainly, the tangent plane of a surface point generally shares only one point with the original object. Thereby we do not take the tangent plane as the candidate for the optimal cutting plane.

According to the definition of the optimal cutting plane, since the sectional slice obtained from the cutting plane which is perpendicular with the direction of the local object has the least slice area, the optimal cutting plane should be perpendicular with the local medial axis of the object. The idea of our method is just similar as the situation that when a carrot is cut the sectional slice obtained by slope-cutting is always bigger than the vertical sectional slice for the same specified surface point.

A sectional slice obtained by cutting a closed surface model approximately likes an ellipse. The principle of our method takes the area of each ellipse as the base to determine the optimal cutting plane. But in practical processing, for scattered point clouds model, the precise closed curve as the boundary of sectional slice is difficult to obtain. In addition, it is imprecise to calculating the area of a sectional slice with a small constant thickness. Alternatively, we calculate the average value of all distances from the center of a sectional slice to each surface point lying on the corresponding cutting plane. Then the average distance is used to substitute the area of the sectional slice. Therefore the sectional slice with minimal average distance belongs to the optimal cutting plane.

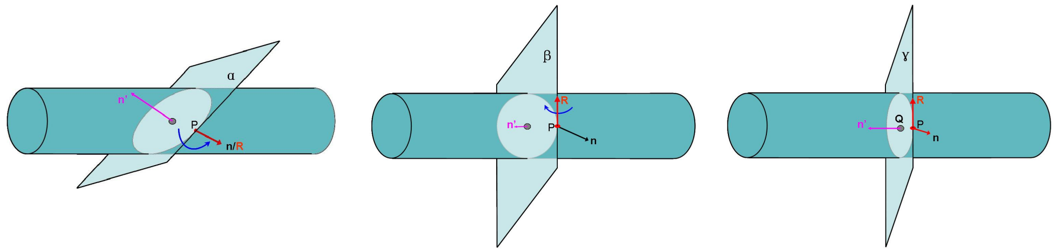
3.1 The optimal cutting plane

The optimal cutting plane can be determined by the next three steps. Fig. 1 illustrates the principle of determining the optimal cutting plane for a surface point.

Step1 Select an initial cutting plane for a specified surface point.

As previous statement, the plane which has the minimal average distance should be selected as the optimal cutting plane for a specified surface point. An exceptional case is that the tangent plane of a surface point mostly has only one common point with the original object. Its average distance becomes zero. Obviously the tangent plane should not be selected as the optimal cutting plane for the surface point.

In order to determine the optimal cutting plane for a specified surface point, the search processing from an initial cutting plane should be started. A family of planes (all planes pass both the specified surface point and its normal vector) must excludes the tangent plane of the specified point. In our method, any one plane selected randomly from this family of planes can be used as the initial cutting plane for the specified surface point. For example, in Fig. 1(a), the plane α passing both point $\mathbf{P}$ and its normal vector $\mathbf{n}$ is selected as the initial cutting plane of point $\mathbf{P}$.



(a) (b) (c)

Fig. 1: The process of finding the optimal cutting plane for a surface point P. (a) Select an initial cutting plane α for a surface point P. (b) Determine the optimal cutting Plane β which passing the normal vector of point P. (c) And the optimal cutting plane γ of point P.

Step2 Determine the optimal cutting plane passing both the surface point and its normal vector.

This step takes the normal vector **n** of the specified point **P** as the rotation axis **R**. It rotates the initial cutting plane α in an identical direction (clockwise or counter-clockwise) with a little constant angle θ around the axis **R** successively. Each time it will produce a sectional slice until this cutting plane has totally rotated 180 degrees. The cutting plane whose sectional slice has the least average distance will be selected as the current optimal cutting plane.

Taking standard cylindrical shapes into account, if the rotation goes along an identical direction, the area of sectional slice would monotonically change before it gets the extremum. Therefore, once the rotation direction corresponding to declining of the average distance is determined, the cutting plane α will need to be rotated no more than 90 degrees totally in deed. Fig. 1(b) shows that the plane β is the optimal cutting plane passing the normal vector of point **P**.

Step3 Obtain the optimal cutting plane just passing the specified surface point.

The optimal cutting plane of a surface point **P** actually does not necessary have to go through the vector **n** (the normal vector of point **P**). Although the preceding two steps have found the optimal cutting plane which passes through both the point **P** and its normal vector **n**, the final optimal cutting plane γ that just passing point **P** still need to be searched by another set of rotations.

The rotational axis should be specified at first. The cross-product of two vectors **n** and **n´** is used as the current rotational axis. Where **n** is the normal vector of the surface point **P** and **n´** denotes the unit normal vector of the cutting plane β. Sequentially, the cutting plane β is rotated with a certain angle along an identical direction and the average distance of sectional slice is recorded each time just as what we have done in step2. Consequently, it will get the minimal average distance which corresponding to the sectional slice of the final optimal cutting plane for point **P**. Fig. 1(c) shows plane γ is the optimal cutting plane associating to point **P**.

Similarly, this step also needs no more than 90 degrees total rotation to gain the final optimal cutting plane γ which is approximately perpendicular with the medial axis of the local object.

A circle-like sectional slice with a small thickness will be obtained when the algorithm finds the optimal cutting plane for a specified point. It takes the center of this sectional slice as the skeleton point. And the correspondence between this skeleton point and those surface points lying on the sectional slice is generated. We store both the distances from these surface points to the skeleton point and the orientation of each surface point related to its skeleton point. After each point on the surface of a cylindrical object has been associated to their sectional slice of the optimal cutting plane, all skeleton points of the whole 3D object can be extracted from these sectional slices.

The curve-skeleton of an object will be produced by connecting all these skeleton points. There are some algorithms in graph theory that can be utilized to reconnect a set of discrete skeleton points, for example, minimum spanning trees [24, 25] and shortest paths [26]. Because all skeleton points are not well organized and without any preprocessing in our method, we use the algorithm of minimum spanning trees to construct the curve-skeleton of the example models.

3.2 Segmentation of multiple regions

Taking the complexity of 3D object into consideration, there will be two (or several) slices in one cutting plane when two (or more than two) branches are cut by a same optimal cutting plane. The center O of all points located on one optimal cutting plane, as shown in Fig. 2, will be taken as the skeleton point which is even outside these two slices. Obviously, all the points on this identical cutting plane are incorrectly considered to belong in the same slice and share a common skeleton point. Thus these points should be divided into two groups.

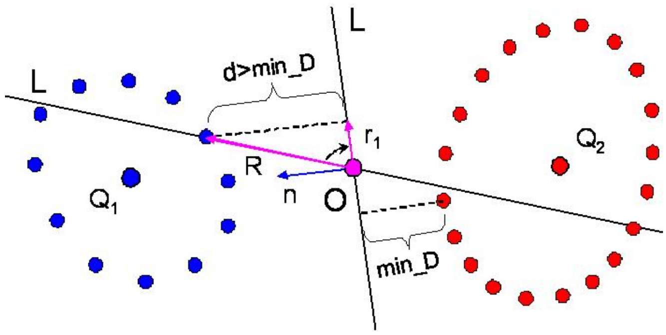


Fig. 2: Segmentation of two regions on one optimal cutting plane.

We can form a line L by connecting any one point located on the plane with point **O**. Then take point **O** as the center and rotate the line **L** with a small angle successively until **L** locates at a position that the distance from each point to the line **L** is greater than a constant minimal distance **min_D**. Then all the points on the same optimal cutting plane are separated by the line **L**. Two skeleton points will be calculated from different groups respectively. In Fig. 2, $\mathbf{Q}_1$ and $\mathbf{Q}_2$ are two skeleton points belong to different shape branches.

3.3 Determine relating parameters

Our method searches the optimal cutting plane by implementing two sets of rotations. The cutting plane is rotated with a constant degree θ in each operation. The value of θ practically affects the result of our algorithm. The cutting plane will miss the optimal position which is approximately perpendicular with the local medial axis of the shape if we choose a large value of θ. However it will take less time to finish the rotation. On the contrary, if θ is selected too small it will not miss the optimal cutting plane but reduce the efficiency of the algorithm. In order to balance this trade-off, we employed a value between 5 and 10 degrees at each time for finding the optimal cutting plane in the examples shown in next section.

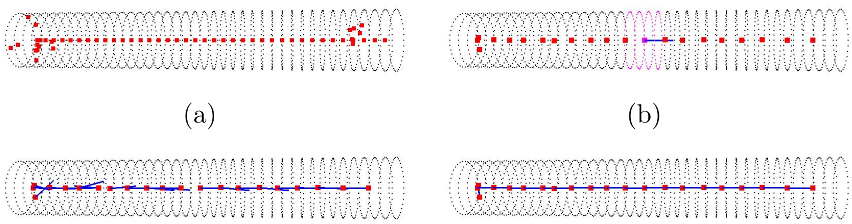
The parameter λ which is the thickness of the sectional slice (also the thickness of each cutting plane) is important to our algorithm. It associates with the density of the original point clouds. Generally, the thinner the cutting plane is, the better quality the curve-skeleton has. According to different application requirements, the value of λ can be chosen differently. Actually λ should be selected neither too small nor too large, otherwise our method could not work properly. In order to control shape deformation for producing computer animation we chose the value of λ as 2 to 10 percentages of the maximal size of the shape.

Another parameter which should be determined is the **min_D**. It is used to segment multiple regions in one optimal cutting plane. However, different objects (even different components of a same object) should use distinct values. Our method will work as long as the positive **min_D** is selected less than the minimal distance from the surface point to the point **O**, as shown in Fig. 2.

4 Experiments of Extracting Shape Skeleton

We use point clouds data of a standard tubular shape to demonstrate the principle of our method. The curve-skeleton of a tube which is generated by our method is showed in Fig. 3. Fig. 3(a) is a point clouds model of a tube. And its skeleton points are extracted from the optimal cutting planes with 5 percents thickness of the maximal shape size. One sectional slice and its orientation are illustrated in Fig. 3(b). Fig. 3(c) shows skeleton points and the orientations of the corresponding optimal cutting planes. The final curve-skeleton of the tube is generated in Fig. 3(d).

From the orientations of the sectional slices we can see that the optimal cutting planes keep approximately perpendicular with the local medial axis of the object and the skeleton points lie on the center of each sectional slice. However, at the ends of the tube it generates some noise, as shown in Fig. 3(a). Especially for complex objects (with many branches) the optimal cutting plane method has to involve too much tedious works to detect the joints and remove the noise-points as many other skeleton extracting approaches.



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

Fig. 3: Point clouds of a tube and its curve-skeleton. (a) Skeleton points created by the optimal cutting plane with 5% thickness of maximal size of the object. (b) A sectional slice (in peachblow) with 10% thickness of the object and its orientations (in blue). (c) Skeleton points and their orientations of the corresponding optimal cutting planes. (b) Curve-skeleton generated by connecting skeleton points.

Two other curve-skeletons which are extracted from a dolphin and a horse are shown in Fig. 4 and Fig. 5 respectively.

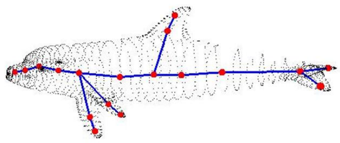


Fig. 4: The point clouds of a dolphin and its curve-skeleton.

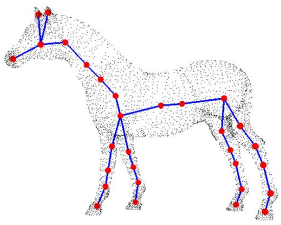


Fig. 5: The point clouds of a horse and its curve-skeleton.

5 Shape Deformation

Shape deformation can be produced based on curve-skeleton of 3D object [8, 9]. In computer graphics and animation fields, users generate different shape postures by following its skeleton deformation.

Exploiting the shape skeletons (generated in section 4), we carried out some transformations (including translations and rotations) on these curve-skeletons and created some new skeleton postures. Then the results of shape deformations can be calculated according to different curve-skeleton postures through the correspondences between skeleton points and its associated surface points.

In Fig. 6, there are some different postures of a horse, which are produced based on their deformed curve-skeletons. Fig. 7 shows another example, a set of deformation results of a dolphin.

6 Conclusion and Future Work

Curve-skeleton is a concise representation of the original object. It benefits many shape manipulations. Although a great deal of work has been done in skeleton extraction from surface shapes and solid models, the method presented in this paper processes point clouds data directly.

Our method based on optimal cutting planes to produce the skeleton points. The optimal cutting plane is obtained by only implementing two sets of rotations. And it keeps approximately perpendicular with the local medial axis of a shape. This property correctly attaches original point clouds to the finial skeleton points. Therefore it contributes to skeleton-driven shape deformation. The experimental results show that the skeletonization method based on the optimal cutting plane is effective for simple cylindrical shapes. Employing curve-skeleton can greatly facilitate the generation of shape deformations.

The skeleton extraction method based on the optimal cutting plane is affected by the quality of point clouds data. It could obtain more higher-quality skeleton points if the point clouds are

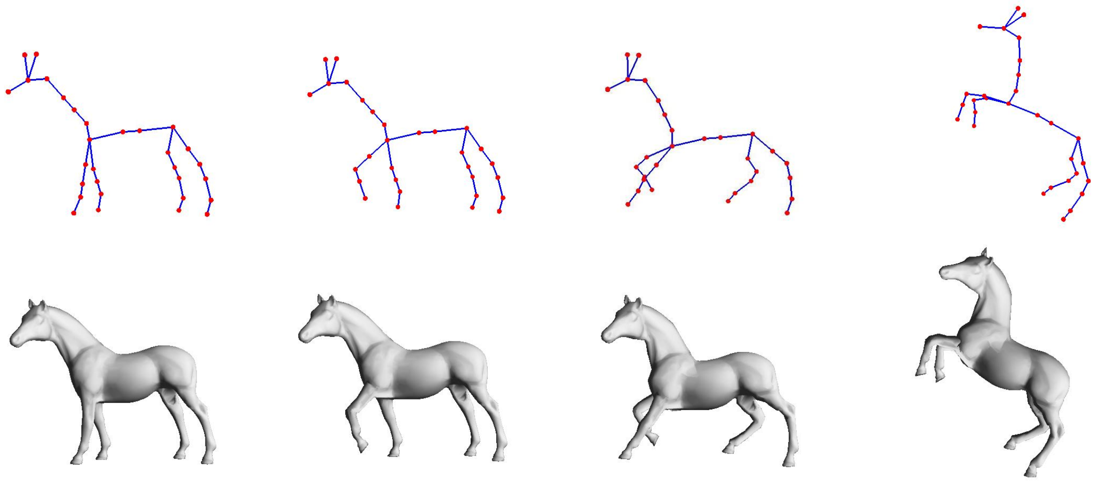


Fig. 6: One set of postures (a horse) generated by its skeleton deformation.

dense enough and distribute evenly. Another factor should be concerned is the complexity of the original object. If the 3D object has more branches (containing many joints) and complex structure, it needs too much pruning manipulations which inevitably reduces the performance of the algorithm. However, our method should be more effective for general cylindrical shapes if we can correctly solve the joints detection and segment the shapes in proper parts before the skeletonization.

For future work, we would like to process the detection of joints more efficiently and try to explore the proper clustering method to segment the initial shape. Furthermore, in order to generate smooth shape gestures, we expect to conquer the overlap in articulated parts when the large scale shape deformation occurs.

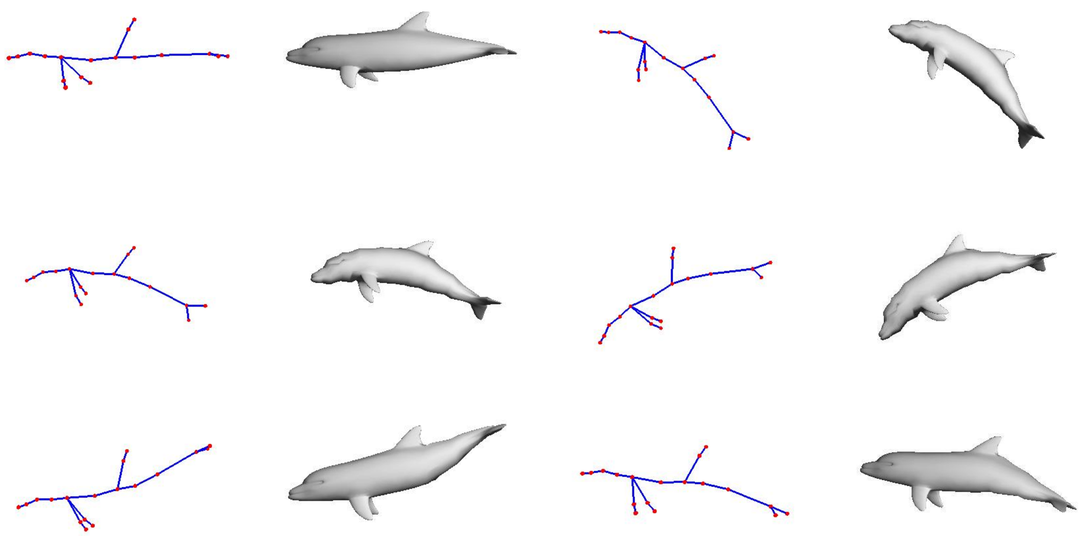


Fig. 7: Another set of postures (a dolphin) created by its skeleton deformation.

References

[1] Blum H.: A transformation for extracting new descriptors of shape. Models for the perception of speech and visual form. MIT Press, 362-380 (1967).

[2] Shotton J., Fitzgibbon A., Cook M., Sharp T., Finocchio M., Moore R., Kipman A. and Blake A.: Real-time human pose recognition in parts from a single depth image. In Proc. CVPR, 2011.

[3] Wei S., Ke D., Xiang B., Tommer L., Baining G. and Zhuowen T.: Exemplar-Based Humman Action Pose Correction and Tagging. In Proc. CVPR, 2012.

[4] Brostow G.L., Essa I., Steedly D. and Kwatra V.: Novel Skeletal Representation for Articulated Creatures. Proc. ECCV, 2004.

[5] Brennecke A., Isenberg T.: 3D Shape Matching Using Skeleton Graphs. Simulation and Visualization, 299-310, 2004.

[6] Hilaga M., Shinagawa Y., Kohmura T., and Kunii T.L.: Topology matching for fully automatic similarity estimation of 3D shapes. Proc. of SIGGRAPH, 203-212, 2001.

[7] Lewis J.P., Cordner M., Fong N.: Pose Space Deformation: A Unified Approach to Shape Interpolation and Skeleton-Driven Deformation. Proceedings of SIGGRAPH on computer graphics and interactive techniques, 2000.

[8] Yan H.B., Hu S.M., Martin R.R. and Yang Y.L.: Shape Deformation Using a Skeleton to Drive Simplex Transformations. IEEE Transactions on Visualization and Computer Graphics. 14, 693-706, 2008.

[9] Bloomenthal J.: Medial Based Vertex Deformation. SIGGRAPH Eurographics Symp. On Computer Animation, 147-151, 2002.

[10] Yu Y., Mao T.L., Xia S.H. and Wang Z.Q.: A Pose-Independent Method of Animating Scanned Human Bodies, Computer Graphics International. 232-239, 2008.

[11] Telea A., Vilanova A.: A Robust Level-Set Algorithm for Centerline Extraction. IEEE Symp. On Data Visualization, 2003.

[12] Cornea N.D., Silver D., and Min P.: Curve-Skeleton Properties, Applications and Algorithms. IEEE Trans. Vis. and Comp. Graphics, 13, 530-548, 2007.

[13] Borgefors G., Nystrom I., Sanniti di Baja G.: Computer skeletons in three dimensions, Pattern Recognition, 32, 1225-1236, 1999.

[14] Borgefors G.: On Digital Distance Transforms in Three Dimensions. Computer Vision and Image Understanding. 64, 368-376, 1996.

[15] Patane G., Spagnuolo M. and Falcidieno B.: Reeb graph computation based on minimal contouring. In Proc. IEEE Conf. on Shape Modeling and App., 73-82, 2008.

[16] Ogniewicz R., Ilg M. and Zurich E.: Voronoi skeletons: theory and applications. In Proc. CVPR, 63-69, 1992.

[17] Katz S., and Tal A.: Hierarchical mesh decomposition using fuzzy clustering and cuts. ACM Trans. on Graph. 22, 954-961, 2003.

[18] Dey T., and Sun J.: Defining and computing curve-skeletons with medial geodesic function. In Symp. on Geom. Proc., 143-152, 2006.

[19] Oscar K.C.A., Chiewl L.T.: Skeleton Extraction by Mesh Contraction. ACM Transaction on Graphics, 27, 44, 1-10, 2008.

[20] Sharf A., Lewiner T., Shamir A. and Kobbelt L.: On-the-fly curve-skeleton computation for 3D shapes. Computer Graphics Forum (Proc. of Eurographics) 26, 323-328, 2007.

[21] Verroust A.,Lazarus F.: Extracting skeletal curves from 3D scattered data. Proceedings of International Conference on Shape Modeling and Applications. 1999.

[22] Lazarus F., Verroust A.: Level Set Diagrams of Polyhedral Objects. Proceedings of ACM symposium on Solid modeling and applications. 130-140, 1999.

[23] Andrea T., Hao Z., Daniel C.O.: Curve Skeleton Extraction from Imcomplete Point Cloud. ACM Transaction on Graphics, 27, 3, 2009.

[24] Sundar H., Silver D., Gagvani N. and Dickinsoon S.: Skeleton Based Shape Matching and Retrieval, Proc. Shape Modeling International, 2003.

[25] Wan M., Dachille F. and Kaufman A.: Distance-field Based Skeletons for Virtual Navigation, IEEE Visualization. 2001.

[26] Wade L., Parent R.E.: Automated generation of control skeletons for use in animation, The Visual Computer. 18, 97-110, 2002.