WYSIWYG: Mesh Decomposition for Static Models

Yu-Shuen Wang, Tong-Yee Lee
Computer Graphics Group/Visual System Lab. National Cheng-Kung University, Tainan, Taiwan, R.O.C.
Email: tonylee@mail.ncku.edu.tw

Abstract

Mesh decomposition plays an important role in computer graphics. However, the requirements for each of these applications demand the mesh decomposition process differently and therefore various decomposition schemes were developed as an end result. In this paper, we propose a flexible "WYSIWYG" (i.e. what you specify is what you get) decomposition algorithm for static meshes.

1. Introduction

Mesh decomposition for various applications has been under intense study for several years. Using different partitioning metrics, many approaches (automatic or semi-automatic) have been proposed [1,2,3,4,5,6,7,8,9,10,11,12,13]. However, it is very difficult to have a universal, automatic decomposition method to satisfy the needs of various applications. The semi-automatic methods, in particular, are very useful to interactive applications such as mesh edition and animation. In this paper, a fast and intuitive "WYSIWYG" mesh decomposition scheme is presented. This new technique avoids under-segmentation and over-segmentation problems in most automatic schemes and also avoid tedious, less-intuitive, sometimes, misleading, user specifications in most semi-automatic methods.

2. System Overview

The idea behind our system framework is to set large distance to the cutting positions on the dual graph of a given mesh. The proposed system is composed of five major stages below:

1. Generating Dual Graph for the mesh and assigning distances to all pairs of faces in the mesh.
2. Finding the feature points by user specification.
3. Assigning each face to a certain partition based on the distance to that feature point.
4. Checking on each partition if all faces in the same partition are connected to each other.
5. Smoothing the cutting boundaries using [1] between the partitioned components.

The details of these five stages will be presented in the following sections.

3. WYSIWYG Approach

3.1 Dual Graph Generation

In order to find the distance from one face to other faces through the surface of the mesh, we have to define the distance metrics between two adjacent faces. After defining this distance metrics, then we can connect each of them by an edge to form the Dual Graph. Finally, we apply all-pairs shortest path algorithm to solve the distance between all faces. Currently, there are two distance metrics used to generate Dual Graph in the system. For static meshes, we refer to [1] and set the distance between two adjacent faces by the combination of angle between them and the geodesic distance from the center of one face to the center of the other. The formal expression is shown as follow:

$$Weight(dual(f_i), dual(f_j)) = \delta \cdot \frac{Geod(f_i, f_j)}{avg(Geod)} + (1-\delta) \cdot \frac{Ang\_Dist(\alpha_{ij})}{avg(Ang\_Dist)},$$

where Geod(f_i,f_j) is the geodesic distance between face i and face j, α_{ij} is the angle between these two faces, and δ is the proportion (weight) of geodesic distance and angular distance which is adjusted by user.

3.2 Feature Specification

In [1], the system finds faces which are far away from each other and set them as feature points automatically. But as we mentioned before, sometimes user may want to separate the particular partition from original mesh, the automatic process can not always satisfy user's desire. Therefore, our system decides to take user specifications as feature points and decomposes the static meshes accordingly.

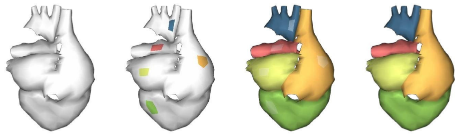

Figure 1: (a) Original mesh. (b) The marked region stands for a partition with certain color. (c) Cutting result with the marked region in high light. (d) Cutting result.

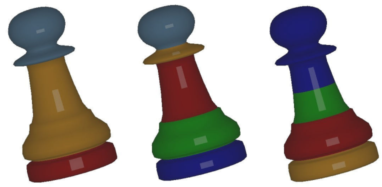

Figure 2: Different results with different marked regions specified by the user.

In the proposed system, the user-specified features will be assumed to be the rough center of the partitions. An illustration of the user specification is shown in Figure 1. A human heart is shown in Figure. 1(a). The user draws some areas on different partitions with different colors as shown in (b). After this specification process, our system will decompose the mesh according to this guide, and the result is shown in (d).

Figure 2 is a chess decomposed with various purposes. High-light area is the marked region through user input device. The mesh is cut at the concave region in principle, but the cut can be located at a flat region as well if user desire to do so. It is obvious to note that the number of partitions and the cutting boundary follows the user specification.

For most cases, only one feature point is needed by one partition; but sometimes it is not enough, especially when those two nearby partitions have no concave edges or deformation similarly. In this situation, the effective distance on the Dual Graph is composed only by geodesic distance. Therefore, the ideal cutting boundary is not clear to our system and may be different to users. To have a better control over the decomposition, the user should point out more features on the mesh to specify the partition. For example, Figure 3 is an almost convex object; the angular distance has slight influence on the cutting boundary. Thus, meaningful partitions can not be defined definitely and are different from user to user. The user has to point out more features to specify the cutting boundary if he or she has special requirement for the decomposition.

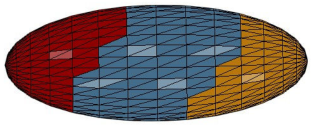

Figure 3: The almost convex object should be decomposed with more feature points to indicate the shape of the boundary.

3.3 Face Clustering

After the position of the feature points are indicated, the system will decide which partition each face belongs to based on the distance between the face and the feature points. Faces are hard to stride across a concave edge because of the distance through it is much larger than a convex one. Therefore, the partitions can be separated at concave edges. To achieve our face clustering, we need to minimize the objective function:

$$\min(\underset{fp \in FP_i}{dis} (f, fp))$$

Here, f is the face which is being considered now, and FP_i is a set of feature points representing partition i. After this process, each face in the mesh can be assigned to a partition. But it is not a stable state. Due to unobvious concave edges or the distance from the face to several feature points are close, some faces near boundary may be assigned to the wrong partition. Therefore, our algorithm execute face clustering as an iterative process, and it will dispatch each face to a partition again and again, until there are no other face transferred from one partition to another.

In the above iterative face clustering procedure, it is a good heuristic to make feature points remaining in the center of the partition. Therefore, in this paper, the objective function we need to minimize is represented as

$$\min(\min_{fp \in FP_i}(dis(f, fp)), \underset{p \in P_i}{avg}(dis(f, p)))$$

Here, P_i is a set of faces belonged to partition i. In the beginning of the process, our algorithm set the subset P_i equal to subset FP_i, and then iterative dispatch each face to a certain partition until it reaches a stable condition.

4. Results

Figure 4 shows two experimental results by using different feature points marked by the user. The user can flexibly decide how many partitions are required and where the cutting boundary should be as long as we draw some region on the mesh. Therefore, it is indeed a helpful “WYSIWYG” mesh decomposition system. Our system can allow a user to decompose a mesh with different hierarchical level, but not limited by it. For example, the arms of Dino can be decomposed into upper arms, forearms, palms and fingers, but other positions are still in the rough level. The left leg of tree frog is decomposed into different levels, but other partitions are not. Figure 5 shows three extra experimental results.

5. Conclusion and Future Work

In this paper, we present a semi-automatic “WYSIWYG” mesh decomposition scheme to let the user flexibly decompose a mesh into several meaningful partitions. Although user specification can decompose a mesh into meaningful partitions for various purposes, sometime it is a little troublesome to point out feature points by user, especially when there are lots of partitions to cut. We believe that advances in artificial intelligence may overcome this situation. The system will learn where the marked regions are depending on the kind of dual graph and result of decomposition when the user draws some area on the mesh. Therefore, the automatic process for future decomposition scheme will be parameter free and satisfy the desire of the user as much as possible.

6. Acknowledgements

This paper is supported by the National Science Council, Taiwan, Republic of China, under contract No. NSC-95-2221-E-006-193-MY2 and NSC 95-2221-E-006-368.

7. References

[1] Sagi Katz and Ayellet Tal, ”Hierarchical Mesh Decomposition using Fuzzy Clustering and Cuts,” ACM Trans. Graph. 22, 3 (SIGGRAPH 2003), 954–961.

[2] A. Shamir, “A Formulation of Boundary Mesh Segmentation,” In *Proc. 2nd International Symposium on 3D Data Processing, Visualization, and Transmission*, September 2004.

[3] Kalvin, A. D., and Taylor, R. H., “Superfaces: Polygonal mesh simplification with bounded error,” IEEE Comput. Graph. Appl. 16, 3, 64–77, 1996.

[4] Garland, M., Willmott, A., and Heckbert, P. S., “Hierarchical face clustering on polygonal surfaces,” In SI3D ’01: Proceedings of the 2001 symposium on Interactive 3D graphics, ACM Press, New York, NY, USA, 49–58, 2001.

[5] Loydll, S., “Least square quantization in pcm,” IEEE Transactions on Information Theory 28, 129–137, 1982.

[6] Chuang, F. R. K., “Spectral graph theory,” CBMS Regional Conference Series in Mathematics, 1997.

[7] Yunjin Lee, Seungyong Lee, Ariel Shamir, Daniel Cohen-Or, and Hans-Peter Seidel, “ Intelligent Mesh Scissoring Using 3D Snakes,” In PG ’04: Proceedings of the Computer Graphics and Applications, 12th Pacific Conference on (PG’04), IEEE Computer Society, Washington, DC, USA, 279–287.

[8] Funkhouser, T., Kazhdan, M., Shilane, P., Min, P., Kiefer, W., Tal, A., Rusinkiewicz, S., Dobkin, D., “Modeling by example,”ACM Trans. Graph. 23, 3, 652–663, 2004.

[9] Lien, J.-M., and Amato, N. M.,“ Approximate convex decomposition,” In SCG ’04: Proceedings of the twentieth annual symposium on Computational geometry, ACM Press, New York, NY, USA, 457–458, 2004.

[10] Katz, S., Leifman, G., and Tal, A., “Mesh segmentation using feature point and core extraction,” Visual Computer 21, 8- 10, 865–875, 2005.

[11] Hsueh-Yi Sean Lin, Hong-Yuan Mark Liao, and Ja-Chen Lin, "Visual Salience-Guided Mesh Decomposition," *IEEE Trans. on Multimedia*, vol. 9, No. 1, pp. 46-57, Jan. 2007.

[12] T.Y. Lee, Y.S. Wang, T.G. Chen, “Segmenting a Deforming Mesh into Near-Rigid Components,” Visual Computer Journal Vol. 22, No. 9-11, Sept. 2006, pp. 729-739 (Pacific Graphics 2006).

[13] Y.S. Wang, T.Y Lee, C.H. Lin, “Interactive Model Decomposition,” to appear in the proceeding of The 10th International Conference on Computer-Aided Design and Computer Graphics (***CAD/Graphics 2007*** sponsored by IEEE and ACM SIGGRAPH), Beijing, China on October 15-18, 2007.

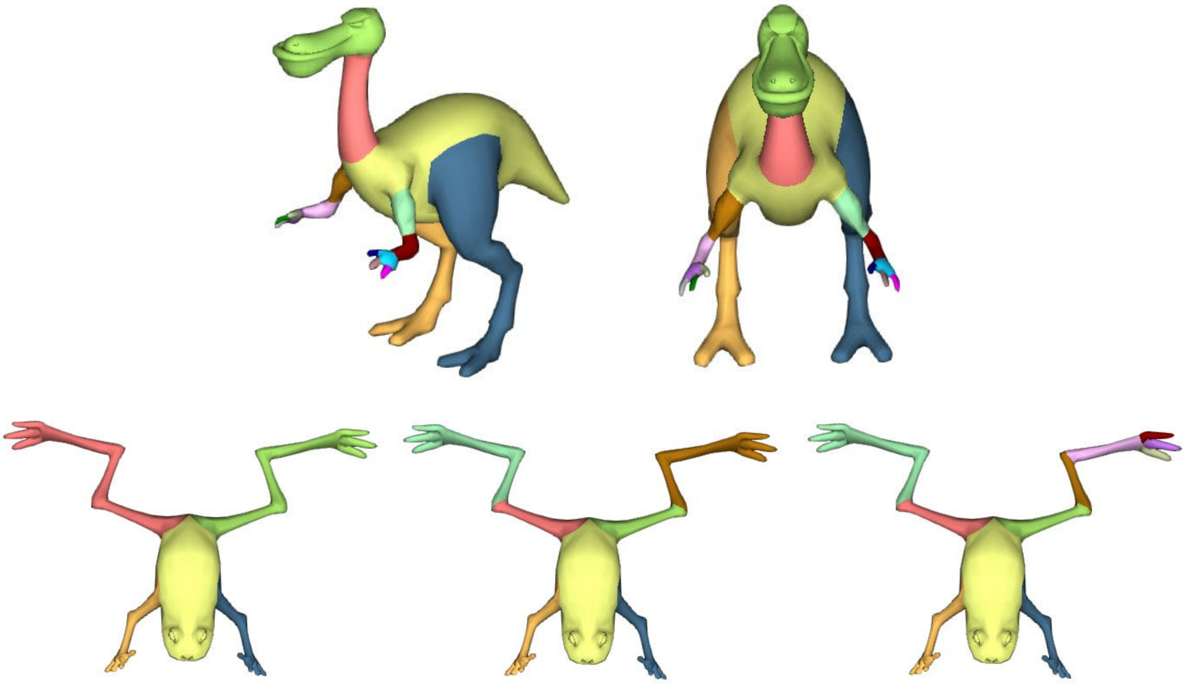

Figure 4: Different hierarchical segmentations of Dino and Tree Frog by user specifications. A user can decide how fine the level is in different region of the mesh.

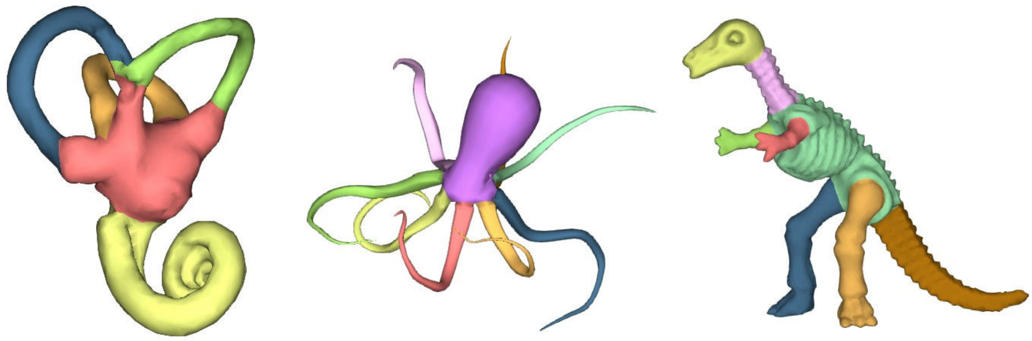

Figure 5: Different hierarchical mesh segmentations by user specifications.