CrossMark

A Survey on Segmentation of 3D Models

Chen He[1] · Chunmeng Wang[2]

Abstract Segmentation of three dimensional models is a basic problem in computer graphics research. In recent years, lots of algorithms have been proposed in this field, providing varieties of methods and evaluation standards. This article gives a comprehensive review of the state-of-the-art of segmentation techniques of 3D models. We define the segmentation problem, classify previous segmentation solutions into different types and describe the typical algorithms according to the time sequence. We also give some trends in segmentation research of 3D models.

Keywords Survey · Segmentation · Partition · 3D model

1 Introduction

Three dimensional (3D) model segmentation is an important problem in many graphics and/or digital geometry processing applications. Segmentation of 3D models can be used in many mesh processing tasks, including texture mapping [1, 2], 3D model compression [3, 4], simplification [5], skeleton extraction [6], 3D shape retrieval [7] and so on. Moreover, a meaningful decomposition of 3D surface meshes can provide semantic information about the underlying objects.

1.1 Definition of 3D Model Segmentation

The most commonly used representations for 3D models are surface meshes and/or point clouds. The definition of the 3D model segmentation will be given in this section, taking meshes for example.

✉ Chen He
imhechen@163.com

[1] Media and Communication College, Weifang University, Weifang, China

[2] Computer Engineering College, Jinling Institute of Technology, Nanjing, China

Commonly, A 3D mesh M can be defined as a three tuple $M = (P, E, F)$. In this definiton, P represents points and $P = \{p_i | p_i \in R^3, 1 \leq i \leq n\}$, E represents edges and $E = \{e_{ij} \mid e_{ij} = (p_i, p_j), p_i, p_j \in P, i \neq j\}$, F represents faces and usually, triangle faces $F = \{f_{ijk} \mid f_{ijk} = (p_i, p_j, p_k), p_i, p_j, p_k \in P, i \neq j, i \neq k, j \neq k\}$, but faces can also be other types of polygons such as quadrangle. Given a 3D model represented as a boundary mesh $M = (P, E, F)$, the 3D model segmentation can be defined as a clustering problem. In this problem, the clustering primitives can either be P, E or F. But the faces F is usually taken as the clustering element, in that if the elements are P or E, the faces whose points or edges are from different clusters will be divided into none segments.

1.2 Segmentation Type and Objectives

The problem of three-dimensional model segmentation could be classified into two categories: *surface patch segmentation* [1–3, 5, 8] and *meaningful segmentation* [9, 10–18]. Surface patch segmentation proceeds in a purely geometric way, using the curvature or the planarity of the surface, to create the patches. Surface patch segmentation is often employed in texture mapping, building map, and geometric image formation. Meaningful segmentation is to divide the object into meaningful parts. For example, the human model can be divided into head, torso, arms and legs. In this sense, meaningful segmentation is senmantic oriented and its goals are rooted in human perception research. When humans try to understand an image, they usually divide the image into smaller parts first, then recognize the objects and shapes. For this reason, meaningful segmentation breaks down the 3D model into segments that generally are the physical portion of the model.

It should be noted that there were some survey papers on this subject. Attene et al. [19] provide some comparative studies of segmentation algorithms and experimental results from several different angles. They only give the result of the algorithms with the code provided to them, so this is not an extensive study. Agathos et al. [20] present the method of 3D mesh segmentation in detail and examined its suitability for CAD models. Shamir [21] present a review of three dimensional shape segmentation techniques. According to optimization method, characteristics used and different subdivision goals, they categorize the previous proposed segmentation algorithms. However, these survey papers are relatively obsolete. Compared with previous survey papers, our work aims to provide a brief review and up-to-date coverage of the segmentation algorithms for 3D models.

In the following sections, we will briefly introduce the typical segmentation algorithms according to the time sequence. In 2009, Chen et al. [22] proposed a benchmark for three dimensional model segmentatino, which is known as *Princeton Segmentation Benchmark*. This benchmark will be introduced in Sect. 3. Taking this benchmark as a time dividing point, we will respectively describe the algorithms before and after the proposing of this benchmark in Sects. 2 and 4, respectively. Then we will conclude and give some trends in the last section.

2 Recent Segmentation Methods

This section will briefly introduce five typical algorithms which are proposed before *Princeton Segmentation Benchmark*.

2.1 Feature Point and Core Extraction

Katz et al. [9] proposed a novel algorithm for mesh segmentation. The algorithm first extract the core of the mesh object using the salient feature points, then segment the object into meaningful parts. There are three key ideas in this method. First, transform the surface vertices into gesture-insensitive representations by multidimensional scaling (MDS). Second, using the MDS representation to extract salient feature points. Third, find the core components. Finally, 3D objects can be divided into meaningful parts using the core and feature points.

2.2 Fitting Primitives

Given a triangular mesh, a hierarchical face clustering algorithm was proposed in Attene et al. [8]. The algorithm based on the fitting of primitives which are belonged to a set of arbitrary shapes. In their essay, planes, spheres, and cylinders were chosen to fit the original graphics. The proposed clustering algorithm is done automatically and a binary tree for clusters is created. At the beginning, a single cluster only contains one triangle. Then in each iteration, consider all the neighboring pairs of clusters, and a new single cluster may be formed from the one that is better approximated by one primitive shape. When calculate the approximation error, the same measure is used for all primitive shapes, so it is functional to choose the most appropriate primitive shape to approximate a single cluster in the binary tree. After the end of the iteration, the object represented as triangle meshes is segmented into surface patches.

2.3 Shape Diameter Function

Based on the "shape diameter function" (SDF), Shapira and Shamir [10] described a 3D model decomposition algorithm. SDF measures the diameter of a shape's volume near a point on the surface. This algorithm calculate the SDF for each face centroid, then proceed in two steps. First, produce a k-dimensional vector for each face using Gaussian mixture model (GMM). This vector indicates the probability of assigning the face to one SDF cluster. Second, minimize the energy function which combines the k-dimensional probability vector from step 1, and refine the segmentation using alpha extension graph cut algorithm. The algorithm proceeds hierarchically for a given number of "partitioning candidates", which determines the output number of segments.

2.4 Randomized Cuts

Golovinskiy and Funkhouser [11] proposed a hierarchical decomposition algorithm that uses a randomized set of minimum cuts to guide the position of the segmentation boundary. They first extract the mesh and then perform the hierarchical processing from top to bottom, beginning from all the faces in a single fragment and then iteratively doing the binary segmentation. For each group, they compute a random set of cuts for each segment, and then they determine for each segment which is the most consistent cut with the other cuts in the randomized set. In this group of candidate cuts, they chose the one that led to the smallest normalized cut cost. The algorithm ends when a certain number specified by the user is reached.

2.5 Random Walks

Lai et al. [12] describes a segmentation procedure which contains two stages. During the first phase, each face F is assigned to a segment whose seed face has the highest probability to reach F through the random walk on the mesh dual graph, then an over segmentation can be calculated. During the second phase, based on the order of the relative length between the intersection and all of the adjacent segments' total perimeter, the segments are hierarchically merged. Hierarchical clustering algorithm ends when the number of segments specified by the user is reached.

3 Princeton Segmentation Benchmark

In 2009, Chen et al. [22] proposed a benchmark which is used to evaluate 3D mesh segmentation algorithms. The bench ark contains a software which is used to analyze 11 geometric attributes of the segmentation and generating four quantitative metrics as the comparison criteria for the segmentation algorithms. The benchmark also includes a dataset containing 4300 manually generated segmentation datasets for analysis of 380 surface meshes which can be classified into 19 object categories.

The contributions of this benchmark are five-fold. First, they describe a procedure for collecting a distribution of mesh segmentations from people around the world. Second, they investigate four metrics (Hamming distance, Rand index, Cut discrepancy and Consistency error) for comparing mesh segmentations, adapting ideas from computer vision to the domain of 3D meshes. Third, they give the result of quantitative comparison of seven typical automatic segmentation algorithms according to these metrics. Fourth, they analyze properties of segmentations generated by humans and computers respectively, making observations about how they are similar and different with respect to one another. Finally, they provide a publicly available data set and software suite for future analysis and comparison of mesh segmentations (http://segeval.cs.princeton.edu).

4 Most Recent Methods

After proposing of the *Princeton Segmentation Benchmark*, more and more algorithms are proposed by the researchers. Some typical algorithms are introduced in the following subsections.

4.1 Learning 3D Mesh Segmentation

Kalogerakis et al. [13] introduced a data driven method to simultaneously divide and mark parts in 3D shapes. The labeling of the mesh segment is expressed as the conditional random field (CRF) optimization problem. This divides the mesh into sections, an give each section a consistent label. The objective function of the CRF contains unitary items and binary items between adjacent face tags. The unitary item evaluates the face-to-label consistency in CRF function. The CRF objective function can be trained from a set of marked testing meshes. Use the Joint Boost classifier to learn the basics of the CRF, which is automatically selected from lots of geometric characteristics to select a classifier associated with a particular segmentation task. Other CRF parameters can be learned by the

holdout validation. Without adjusting any parameters manually, the user can specify different segmentation tasks by providing the examples for new tasks. Once learned, the algorithm can be used to automatically segment and label the objects with the same type.

4.2 Heat Mapping

Based on the heat mapping, Fang et al. [14] proposed a new method for 3D mesh segmentation, which is called perceptually consistent mesh segmentation (PCMS). They develop a stable PCMS solution using heat intelligence as a structural awareness message on the surface of a mesh object. PCMS scheme proceeds in three steps. First, estimate the segmentation number by analyzing the Laplace spectrum's behavior. Second, find the most characteristic vertex on every part, which is called heat center, by a heat center searching algorithm proposed in this paper. Third, the heat-centric decomposition approach has revealed that PCMS is highly consistent with human perception. The internal interpretation of the structure can divide the object into multiple meaningful parts in some degree is close to human perception. PCMS scheme also facilitates the interpretation of 3D mesh segmentation through purely geometric sense or semantic information.

4.3 Joint Shape Segmentation

Huang et al. [15] proposed a shape segmentation method for joint analysis of shape database. This method optimizes the segmentation results on all shapes and subsection correspondence among similar shapes. Joint shape segmentation process the shapes in three stages. First of all, for each shape, they calculate a set of initial segments which can form possible segmentations for a shape. Secondly, for each pair of input shapes, joint shape segmentation is performed to identify similar shapes. Thirdly, multiple joint shape segmentation is performed by optimizing the mapping between all similar shapes. By combining multiple shapes, the method can identify functional components though there are loss of some geometric information in a certain shape.

4.4 Projective Analysis

Wang et al. [17] introduced a novel projection analysis method for semantic partition and labeling of three-dimensional shapes. Given an input three dimensional shape, a collection of 2D projection images are obtained from multiple views. Then each projection image is compared with images in a database to find the most similar ones, using an area-based 2D shape matching algorithm. In this algorithm, Hausdorff distance is used to perceive topological structure by considering the internal holes in the binary images. After finding the similar images in the database, the labels of these images are transferred to the projection image, then the projection image can be segmented. In addition, considering the different view and nonuniform scaling of the object, the projective images are warped by a linear transform. At last, the labels of the projected images from different view are back projected onto the three-dimensional input shape and integrated to get the segmentation results and semantic labels.

4.5 P-Spectral Clustering

Chahhou et al. [23] proposed a new method to get the best 3D mesh segmentation because humans can sense using minimum rules and spectral clustering. The main idea is to encode the structural and geometric information in a single matrix to force the incision to appear in the concave area. Then, the optimal cut on the convex region can be found by completely unsupervised spectral clustering method. Since the algorithm is recursive, it can obtain hierarchical segmentation of the mesh. As a result, users can select a level of detail (LOD) and no further calculations are needed.

4.6 Multi-view Recurrent Neural Network

Based on the multi-view recurrent neural network (MV-RNN), Le et al. [24] introduce a new approach for 3D model segmentation. The architecture combines the convolutional neural networks (CNN) and a two-layer long short term memory (LSTM) to yield coherent segmentation of 3D shapes.

For an input 3D model, the goal is to segment it into parts based on the prior knowledge learned from a pre-segmented training dataset. They design a MV-RNN network to this end. The network takes as input a set of images from multiple views which are equally distributed over the 3D model; segments these images by generating per-view boundary probability maps; correlates them by a two-layer LSTM followed by a fully connected layer and returns the consistent edges which are back projected to the 3D surface and finally integrated by a conditional random field (CRF).

4.7 Heterogeneous Graphs

Theologou et al. [25] presented a completely automatic mesh object segmentation algorithm using heterogeneous graph. First, construct a heterogeneous graph that contains the geometric relationships at face level and patch level. Here, face means a single triangle or polygon in the mesh, while patch means a fragment created by the over-segmentation at the beginning. Heterogeneous graphs are composed of two different graphs: mesh dual graph that uses concave information of the mesh and a graph that represents the geometric relationships of the patches. Segmentation is obtained by analyzing the eigenvectors of the Laplacian operator. Each eigen-vector is processed separately, and the segmentation is performed iteratively. Starting from the tirst eigen-vector whose eigenvalue is nonzero, one segment is extracted for each extremum and added in the segments. When dealing with each eigenvector, the results of the previous eigenvectors and the theory of node set and node domain are also considered. The final segmentation result is obtained by analyzing all eigenvectors, and the number of eigenvectors is derived from the eigen-gap standard.

5 Conclusion and Trends

This paper presents a comprehensive review of recently proposed algorithms for 3D model segmentation. We formulate the segmentation problem, classify segmentation algorithms into two types and describe the typical algorithms according to the time sequence.

From these algorithms, we can see some new trends in this research area. First, meaningful segmentation is getting more attention in this research community. Second,

classical shape segmentation techniques analyze geometric structure of individual shape while recent approaches jointly analyze a database of 3D shapes. Third, data-driven methods are receiving increasing attention for 3D model segmentation, i.e., the machine learning approach is being introduced into this area.

References

1. Sander, P. V., Snyder, J., & Gortler, S. J., et al. (2001). Texture mapping progressive meshes. In *Proceedings of the 28th annual conference on Computer graphics and interactive techniques* (pp. 409–416). ACM.
2. Sorkine, O., Cohen-Or, D., & Goldenthal, R., et al. (2002). Bounded-distortion piecewise mesh parameterization. In *Proceedings of the conference on visualization'02* (pp. 355–362). IEEE Computer Society.
3. Karni, Z., & Gotsman, C. (2000). Spectral compression of mesh geometry. In *Proceedings of the 27th annual conference on computer graphics and interactive techniques* (pp. 279–286). ACM Press/Addison-Wesley Publishing Co.
4. Cheng, Z. Q., Liu, H. F., & Jin, S. Y. (2007). The progressive mesh compression based on meaningful segmentation. *The Visual Computer, 23*(9–11), 651–660.
5. Cohen-Steiner, D., Alliez, P., & Desbrun, M. (2004). Variational shape approximation. *ACM Transactions on Graphics (TOG), 23*(3), 905–914.
6. Katz, S., & Tal, A. (2003). Hierarchical mesh decomposition using fuzzy clustering and cuts. *International Conference on Intelligent Computing Systems, 22*(3), 954–961.
7. Zuckerberger, E., Tal, A., & Shlafman, S. (2002). Polyhedral surface decomposition with applications. *Computers & Graphics, 26*(5), 733–743.
8. Attene, M., Falcidieno, B., & Spagnuolo, M. (2006). Hierarchical mesh segmentation based on fitting primitives. *The Visual Computer, 22*(3), 181–193.
9. Katz, S., Leifman, G., & Tal, A. (2005). Mesh segmentation using feature point and core extraction. *The Visual Computer, 21*(8–10), 649–658.
10. Shapira, L., Shamir, A., & Cohen-Or, D. (2008). Consistent mesh partitioning and skeletonisation using the shape diameter function. *The Visual Computer, 24*(4), 249–259.
11. Golovinskiy, A., & Funkhouser, T. (2008). Randomized cuts for 3D mesh analysis. *ACM Transactions on Graphics (TOG), 27*(5), 145–152.
12. Lai, Y. K., Hu, S. M., Martin, R. R., et al. (2008). Fast mesh segmentation using random walks. In *Proceedings of the 2008 ACM symposium on solid and physical modeling* (pp. 183–191). ACM.
13. Kalogerakis, E., Hertzmann, A., & Singh, K. (2010). Learning 3D mesh segmentation and labeling. *ACM Transactions on Graphics (TOG), 29*(4), 102–110.
14. Fang, Y., Sun, M., Kim, M., et al. (2011). Heat-mapping: A robust approach toward perceptually consistent mesh segmentation. In *Computer vision and pattern recognition (CVPR), 2011 IEEE conference on IEEE* (pp. 2145–2152).
15. Huang, Q., Koltun, V., & Guibas, L. (2011). Joint shape segmentation with linear programming. *ACM Transactions on Graphics (TOG), 30*(6), 125–136.
16. Zhang, J., Zheng, J., Wu, C., et al. (2012). Variational mesh decomposition. *ACM Transactions on Graphics (TOG), 31*(3), 21–32.
17. Wang, Y., Gong, M., Wang, T., et al. (2013). Projective analysis for 3D shape segmentation. *ACM Transactions on Graphics (TOG), 32*(6), 192–202.
18. Chen, H. K., & He, Y. D. (2016). A novel part-salience-based approach to fast iterative 3D mesh segmentation. In *International symposium on computer, consumer and control* (pp. 311–314).
19. Attene, M., Katz S., Mortara, M., et al. (2006). Mesh segmentation—A comparative study. In *Shape modeling and applications, SMI 2006. IEEE international conference on IEEE* (pp. 7–18).
20. Agathos, A., Pratikakis, I., Perantonis, S., et al. (2007). 3D mesh segmentation methodologies for CAD applications. *Computer-Aided Design and Applications, 4*(6), 827–841.

21. Shamir, A. (2008). A survey on mesh segmentation techniques. *Computer Graphics Forum, 27*(6), 1539–1556.
22. Chen, X., Golovinskiy, A., & Funkhouser, T. (2009). A benchmark for 3D mesh segmentation. *ACM Transactions on Graphics (TOG), 28*(3), 73–84.
23. Chahhou, M., Moumoun, L., El, F. M., et al. (2014). Segmentation of 3D meshes using p-spectral clustering. *IEEE Transactions on Pattern Analysis and Machine Intelligence, 36*(8), 1687–1693.
24. Le, T., Bui, G., & Duan, Y. (2017). A multi-view recurrent neural network for 3D mesh segmentation. *Computers & Graphics, 66,* 103–112.
25. Theologou, P., Pratikakis, I., & Theoharis, T. (2017). Unsupervised spectral mesh segmentation driven by heterogeneous graphs. *IEEE Transactions on Pattern Analysis and Machine Intelligence, 39*(2), 397–410.

Chen He is currently a Lecturer at Media and Communication College, Weifang University, China. He received his Ph.D. degree in the School of Computer Science and Technology, Shandong University, in 2014. His research interests include computer graphics, digital image processing, digital geometry processing and data compression.

Chunmeng Wang is currently a Lecturer at Computer Engineering College, Jinling Institute of Technology, Nanjing, China. He received his Ph.D. degree in the lab of human computer interaction and virtual reality from Shandong University in 2015. His research interests include computer graphics, image processing, high dynamic range image.