Compact line-based data structure for region-growing mesh segmentation

Bo Yu \(^1\), Maria Savchenko \(^1\), and Ichiro Hagiwara \(^1\)

\(^1\)MIMS, Meiji University, Japan

Abstract

In the presented paper, a line-based region-growing technique was developed for partitioning the polygonal meshes. This approach is realized through developing a simplified stack (basic computer science data structure) that is called a line-based structure. The line-based structure holds information about only the starting and ending triangles in the generated triangle lines according to the developed algorithm. This structure allows decreasing the memory storage space during the segmentation process and improving the computation efficiency considerably.

Keywords- mesh segmentation, region growing, data structure

1 Introduction

Mesh segmentation has become an important step in model understanding and can be used as a useful tool for different applications, for instance, modeling and reverse engineering. Part decomposition gains attraction since it simplifies the problem with multi-parts, complex objects into several sub problems each dealing with their constituent single, much simpler parts. A number of mesh segmentation techniques have been proposed recently. In [1] the survey and definition of generic algorithms for the major segmentation techniques are introduced. Existing methods are handling triangular meshes segment models into surfaces instead of into meaningful parts. In [2] an exhaustive overview of the 3D mesh segmentation methodologies examining their suitability for CAD models is presented. In the paper, a categorization of the existing 3D mesh segmentation methods is proposed and the basic conclusions about different methods are drawn. The authors also present criteria and features used for each segmentation method. The paper [3] provides a comparative study of the latest algorithms and evaluation results. The authors conclude that each algorithm has benefits and drawbacks and future research works on mesh segmentation will be useful. In the paper [4], a new curvature based algorithm which segments the mesh into several regions is described. In this paper, we suggest an approach for the 3D region growing mesh segmentation based on a line structure that presents a simplified stack, which is connected to the computational time. Our goal is to reduce the computational time of the region growing mesh segmentation.

2 Region-growing method

Region-growing segmentation is a widely used image segmentation algorithm proposed by Adams et al. [5]. The image is segmented with respect to a set of the start points, which also been called seed point. Color differences between the seed points and their neighbors are evaluated by threshold to define if the neighbor points can be allocated to same region with the selected seed point.

Seed point defines the initial location of each target region. Growing order selects the next point to be compared around the seed point. In this paper, an automatic seed selection method is developed based on the idea of image region growing segmentation algorithm with thresholding proposed in [6].

2.1 Seed point selection for region growing algorithm

The seed points can be selected manually or automatically on the target model. Different seed point selection method can lead to different result. Manually selection depends heavily on personal’s judge. So it is difficult to guarantee the accuracy if the target model shape is complex.

In image processing field, there are many researches about making seed point selection automatically. Rolf Adams and Leanne Bischof [7] used an area instead of single point as seed selection result. This method can avoid the influence of noise existed in the original image. Based on the idea in [6], we have developed an algorithm to define the seed point by pre-processing the mesh model before region growing process. The method include following steps: firstly, detect the features (ridges and corners) in the model with triangle normal’s comparison; secondly, select a random potential seed cell in a distance to the feature. And last, during the growing process, for comparison between mesh cells, we use the angle differences between the normal of mesh cells and the given threshold value (in this paper, we use this value as 10º for the initialization), and this value should be changed during the process.

2.2 Stack data structure

After seed selection, the neighboring cells around seed cells should be compared with the seed cell in sequence. If the comparison result fulfills the threshold we set before,
the neighbor cell will added to target region. A well-defined stack data structure should be used to reduce the space consumption during this process. Stack is a typical data structure with data manipulates operations that include initialize, push and pop. Before using the stack, a region of memory to store the stack contents should be allocated. The memory size is proportional to the maximum depth of the stack and the data contents of the stack. In the traditional image region-growing segmentation method, the data content in the stack is the pixel information. While in 3D mesh segmentation based on region growing, the data content is the normal direction of each mesh cell. In realistic, complex objects usually represented by a large amount of triangles, if we do not consider the growing order, the region-growing segmentation will require a very huge amount of memory and time consumption. A line-based data structure [8] is proposed to reduce the maximum stack depth during the region growing process in image processing field.

2.3 The proposed algorithm

In the proposed segmentation algorithm, the growing process is produced according to the topological connections between a seed triangle and triangles that are neighbors to it. The stack data content for the region growing process is the “Zigzag” line data structure. The seed triangle is selected by method described in section 2.2. After the selection of seed triangle, a line structure is generated from the seed triangle. The line triangle is generated from the seed cell by the “zigzag” rule in the following steps: Suppose there are $n$ triangle cells in a triangle line, $n$-1 direction vectors are generated by connecting the center point of the neighboring triangles in the triangle line. After that the direction change values between each pair of connected direction vector are calculated, if the change values are different for each connected pair of direction vector. We define this triangle line as a line structure.

$$V(T_{a1}, T_{b1}) - V(T_{a2}, T_{b2}) = V(T_{a3}, T_{b3}) - V(T_{a4}, T_{b4}) < 0$$

where $V(T_{a1}, T_{b1})$ is the normal direction between two connected triangles $T_{a1}$ and $T_{b1}$. In (1), $<T_{a2}, T_{b2}, T_{a3}, T_{b3}>$ are four connected triangles that used to calculate the direction vectors.

After a “Zigzag” line is generated, the end points of each line structure are stored to the stack during the region growing process. And the neighboring lines for region growing process are detected according to these end points. The time complexity of proposed algorithm is $O(n)$, where $n$ is the element number of the mesh model. And the coefficient for the time complexity is proportional to the stack operation times.

3. Example

To validate our algorithm, we test our proposed line-based region-growing algorithm on several mesh models with different triangle numbers and results are demonstrated in Figure 1.

![Figure 1](image)

- a) Car front (42991 faces, 2360 bytes, 2 sec)
- b) Fandisk (12946 faces, 600 bytes, 0.21 sec)
- c) Chamfer (5600 faces, 380 bytes, 0.16 sec)
- three other models are not demonstrated (Bunny, 69451 faces, 3.3 sec; Detail, 120038 faces, 6.4 sec; Isis, 375736 faces, 26 sec)

4. Conclusion

The time complexity of the commonly used seeded based region growing methods is $O(n^s + S + \log(n))$, where $n$ is the element number and $S$ is number of segments. By using the proposed data structure, region growing is performed in a time liner to the mesh model size $O(n)$.

By examining the running time and memory consumption of the region-growing algorithm, a compact line-based data structure for region-growing segmentation is proposed. The method can reduce the stack operation times during the region-growing process. By testing on several examples, this method can significant improve the speed of segmentation process and reduce memory consumption compared with traditional region growing process.

Automatic seed point selection and the threshold selection should be improved to avoid the parameter dependency for the region growing segmentation algorithm.

References