ISSN 1000-9825, CODEN RUXUEW
Journal of Software, Vol.21, Supplement, December 2010, pp.183−193

E-mail: jos@iscas.ac.cn
http://www.jos.org.cn
Tel/Fax: +86-10-62562563

三维网格的边界强度分割算法*

孙晓鹏[1,2+], 纪燕杰[1], 魏小鹏[2,3]

[1](辽宁师范大学 计算机与信息技术学院,辽宁 大连 116029)
[2](大连理工大学 机械工程学院,辽宁 大连 116024)
[3](辽宁省先进设计与智能计算省部共建教育部重点实验室(大连大学),辽宁 大连 116622)

3D Mesh Segmentation Based on Boundary Strength

SUN Xiao-Peng[1,2+], JI Yan-Jie[1], WEI Xiao-Peng[2,3]

[1](Department of Computer and Information Technology, Liaoning Normal University, Dalian 116029, China)
[2](School of Mechanical and Engineering, Dalian University of Technology, Dalian 116024, China)
[3](Key Laboratory of Advanced Design and Intelligent Computing (Dalian University), Ministry of Education, Dalian 116622, China)
+ Corresponding author: szx@nju.edu.cn, http://cs.nju.edu.cn/szx

Sun XP, Ji YJ, Wei XP. 3D mesh segmentation based on boundary strength. ***Journal of Software*, 2010,21(Suppl.):183−193.** http://www.jos.org.cn/1000-9825/10020.htm

Abstract: In this paper, we present a novel and fast algorithm for 3D meshes segmentation based on the geometric information of boundary. Firstly, according to the topological relationship of original mesh faces, we construct the dual graph, and set the weight of vertex and edge on the basis of geometric information of mesh faces. Using the k-way multilevel cutting method, we segment the dual graph quickly to obtain the pre-segmentation and initial boundary for each patch. Then we define the boundary strength function and the feature boundary of each patch, to construct a deformation contour on the patch's boundary. By minimizing the energy of the deformation contour, the initial boundary is driven to approach the feature boundary. Finally, the original mesh is segmented into several meaningful patches in accordance with the minimal rule. The experiment results suggest that our algorithm is efficient and effective, and is applicable to a variety of triangular mesh models with prominent shape features.
Key words: 3D mesh segmentation; boundary strength; deformation contour

摘 要: 提出一种新的基于网格边界几何信息的快速分割算法,首先按照原始网格模型面片的拓扑关系建立对偶图,并根据网格面片的几何信息设定顶点权和边权;使用 *k*-way 多级分割方法在对偶图上进行快速分割,得到预分割区域以及各分割区域的初始边界;然后定义分割片的特征边界和边界强度函数,用以表示各预分割区域边界上的形变模型;通过最小化形变模型的能量函数,推动初始边界向特征边界运动,最终得到符合最小值法则的有意义的子网格.实验结果表明,该算法快速有效,适用于各种局部边缘特点较显著的三角网格模型.
关键词: 三维网格分割;边界强度;形变模型

1 Introduction

Mesh segmentation (or mesh partitioning) has become a fundamental research in digital geometry processing.

* Supported by the National Natural Science Foundation of China under Grant Nos.60873110, 60875046; the Foundation of Department of Education of Liaoning Province of China under Grant No.L2010231

Received 2010-07-20; Accepted 2010-11-03

According to the states-of-the-art, 3D segmentation is also widely applied in a diverse set of model analysis and understanding domain including: modeling[1], morphing[2], collision detection[3], skeleton-driven animation[4], reverse engineering of CAD models[5] and so on.

Differ to image segmentation, the 3D mesh model has extremely higher random in its low level geometry and topology[6]. Most of 3D mesh segmentation algorithm is designed basing on image segmentation, such as watershed segmentation proposed by Mangan and Whitake[7], feature point and core extraction algorithm proposed by Katz[8], local slippage analysis and multi-channel region growing method proposed by Gelfand and Guibas[9], and random walks proposed by Lai[10], but till now, there is not a general algorithm being suitable for all 3D mesh models till now. All those existing algorithms were proposed to aim at different goals of application.

This paper presents a 3D mesh model segmentation algorithm using boundary strength, which is based on the function of boundary strength used in image segmentation.

2 Related Work

The boundary strength method of mesh is borrowed from image segmentation. In image segmentation, this method classifies pixels into several initial small regions according to the gray level and the value of local feature. Then, the difference of values between the neighboring regions is defined as the function of boundary strength, for the merging of the trivial regions which is produced by over segmentation[11]. This method has been proved to be quite useful in capturing essential shape feature of the target area effectively, and be applied to object recognition successfully[12]. One of its special properties is that it encodes the local homogeneous information. This makes it possible to integrate boundary and local homogeneous information, and to be propitious to design shape energy function which has equivalence conformation[13]. It is possible to change the shape energy by adding a weight simply that is proportional to local homogeneous strength, and does not require embedded target to be a closed curves.

Mumford and Shah first proposed the minimizing energy model to formulate the problem of image segmentation as a problem of functional minimum[14]. However, the piecewise smooth approximation of image and the recovering of a set of discontinuity locus corresponding to object boundaries, made it more difficult to minimize the energy function. By defining different function value between smooth and non-smooth boundary, Ambrosio-Tortorelli proposed a guidance function on the smooth boundary to figure out this problem[15]. E. Erdem modified Ambrosio-Tortorelli function, using the boundary strength function as the smooth guide function which embedded in the boundary[16,17]; the algorithm can describe the energy function of the boundary, and obtain prior guide shape by minimizing the energy function. The advantage is that it's able to deal with the morphological variation by the transformation in Euclidean plane, and does not need to construct a simple closed curve for the guiding model. T. Chan proposed a novel active contour model to detect given images[18]. To obtain the feature boundary, this model constructed a mean curvature flow of a dynamic active contour, instead to define the boundaries gradient operator. The initial curve of the model can start from any position of the image, and can detect the internal contour automatically.

In this paper, we investigate a novel mesh segmentation method basing on the energy minimum model[14] and the boundary strength function[16,17]. The basic idea is to define a novel function of boundary strength for the pre-segmentation patches, then construct strain deformation contours, guide the contour to move towards the feature boundary to make the segmentation more meaningful.

Comparing with the guide function in Ref. [15], this paper makes the follow research contributions. First, when we construct the deformation contour, it is not necessary to sect the initial boundary according to its smoothness.

Second, our boundary strength function can ensure the smoothness of boundaries during deformation contour move to the feature boundary, and avoid over segmentation.

3 Overview of Approach

Our strategy is motivated by the following facts: the segmentation method based on regional feature (such as clustering methods) leads to over-segmentation problem easily, while the method based on the boundary dividing line produces incomplete boundary, and repair is needed before segmentation with very larger computation, those algorithms are not suitable for large-scale mesh model.

So, we combine the k-way multilevel partitioning and the algorithm based on the boundary cutting line. The k-way multilevel cutting method is applied to mesh as a pre-segmentation to generate a set of patches with closed boundary, and then detect the feature boundary of mesh on the information of curvature. Construct the deformation contours and define the strength function on the patches boundary, move the deformation contours move to the feature boundary with the impetus of the boundary strength function. When the function obtains its minimum value, we obtain the final boundary of post-segmentation on the position of deformation contour. The input to our processing pipeline is a 3D surface mesh M, and the processing pipeline proceeds in four main steps:

Step 1: Based on the curvature of the vertex on triangle faces of M and the dihedral angle between one triangle face and its neighbor, we define weights ω_f and ω_d. Then pre-segment M into several patches using k-way multistage segmentation method based on ultra large-scale mesh. Denote the patches boundary as initial boundary, which is not ideal, and will be optimized during post-segmentation described as step 2~4.

Step 2: For the model with patches boundary needing optimization, we calculate its average curvature histogram, and then define curvature threshold value by the histogram rugged topography analysis method to classify the vertices of M. The feature boundary is defined by the vertices on the cutting edges.

Step 3: Define the boundary strength function and the deformation contour on each initial boundary. The boundary strength function including two parts, the internal outline impetus strength function Einternal and feature contour restraint strength function Eexternal.

Step 4: Minimizing the boundary strength function, the deformation contour moving from the initial boundary of the patch, is impeded to move towards the feature boundary. When the function obtains its minimum value, we obtain the final cutting boundary of post-segmentation at the position of the deformation contour

4 Pre-Segmentation

The iterative clustering and hierarchical clustering are popular segmentation algorithms too. In the context of surface meshes, typical cost functions include the minimum cut cost of any edge between two patches, and total cost of the cut between two patches, and the Normalized Cut cost[19]. Shlafman[20] used k-means cluster method to segment the mesh model into meaningful parts, and Katz[4] segment the mesh model from the top based on clustering natural level characteristic division. The two kinds cluster methods' shortcoming is that they cannot obtain boundary precisely because of the over segmentation and the boundary jaggy.

We benefit the key idea of the clustering in our pre-segmentation process, to obtain a rapid and effective segmentation, which cluster the face with same geometric property of M into an identical patch, and leave the jaggy boundary to post-segmentation: we construct the dual graph M_d of M based on its topology relations of every triangle face, and define the dual-distance for each edge on M_d, then, cut M_d with Multilevel k-way partitioning scheme. After the pre-segmentation, the boundary of every patch is denoted as the initial boundary.

4.1 Weights computation

Choice of suitable weights assignments for each edge on M_d is essential for our approach to measure the concave-convex features of M, and to give good segmentation results. Appropriate weights are affected by the types of models, due to the different purposes of application. The most important information for segmentation comes from the minima rule, where significant concave features are considered as important hints. We define the weight of each edge on M_d on the combining of curvature information[21,22] and the dihedral angle of adjacent patches[4].

We denote K_i as the Gaussian curvature of each vertex v_i on M, H_i as its Mean curvature, and k_1, k_2 as two principal curvatures, then the Gaussian curvature K_{id} can be defined as the product of the two principal curvatures $K_i=k_1\times k_2$, and the mean curvature H_i can be defined as the average of the two principal curvatures: $H_i=(k_1+k_2)/2$. Then, for every triangles face f on M, its curvature tensors on three vertices v_i (i=1, 2, 3) can be denoted as C_i (i=1, 2, 3), where $C_i=0.5|K_i|+0.5(H_i)^2$. Then we define a curvature weight ω_f for every triangles face f on the mesh model M as follows:

$$\omega_f=\frac{1}{3(C_1+C_2+C_3)}.$$

For two given adjacent faces f_i and f_j, the distance between them is defined as:

$$d(f_i, f_j)=\eta[1-\cos(\text{dihedral}(f_i, f_j))]=\eta/2\|N_i-N_j\|.$$

where dihedral(f_i, f_j) is the dihedral angles between f_i and f_j, N_i, N_j is the normal vector of f_i and f_j, and coefficient η is used to define the bump of dihedral angle, when $\eta>1$, it expresses concave dihedral angle, and when $\eta<1$, it expresses convex dihedral angle. So we have the distance weight ω_d denoted as following:

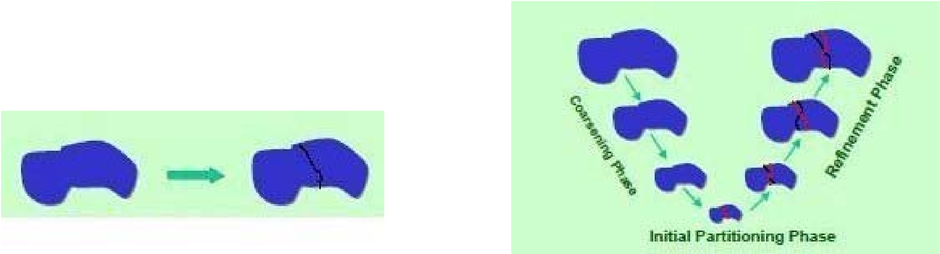
$$\omega_d=|(\text{Centre}f_i, \text{Centre}f_j)|\times d(f_i, f_j)=\frac{\eta\times|(\text{Centre}f_i, \text{Centre}f_j)|}{2\|N_i-N_j\|},$$

where |Centre f_i, Centre f_j| is the sum of two distance between the midpoint of common edge and the center of f_i and f_j respectively.

4.2 Pre-Segmentation with *k*-way partition

Aimed at the problem of the large scale mesh segmentation, Amine et al proposed several parallel *k*-way partitioning algorithm basing on multi-level graph segmentation method[23−25]. Those algorithm reduce the size of the graph by collapsing vertices or edges, until the scale of layers are small enough, then partition the smaller graph, and un-coarse it to construct a partition for the original graph. This algorithm provides a fast and high-quality segmentation for large-scale irregular mesh.

As in Fig.1, multilevel partitioning is divided into three phases: coarsening phase, initial partitioning phase and refinement phase. (1) During the coarsening phase, a sequence of scale decreased graphs is constructed to approach to the original. Obtain graph of smaller scale by collapsing vertices and edges of former hyper graph. It is easier to find a high-quality partition at the coarsest graph. (2) During the initial partitioning phase, use a balanced *k*-way method to split the small graph, as a result, the partition is on the graph whose scale is much smaller than the original hyper graph. Therefore, the divided process can be in a very short period of time to get results. (3) During the refinement phase, partitioning on the coarse layer of hyper graph is mapped to the smaller hyper graph by layer, which achieved the segmentation on the last level of hyper graph, Then optimize the adjustment of the segmentation results after the mapping.



(a) Compute partition directly on the original graph by traditional partitioning algorithms

(b) Compute a partition at the coarsest graph by multilevel partitioning algorithms and then refine the solution

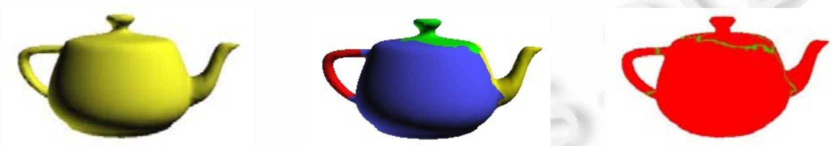
Fig.1 Traditional partitioning algorithms and multilevel partitioning algorithms

For a vertex v_0 in M_d whose corresponding face on M is f, and the k vertices adjacent to v_0 in M_d refer to v_1, $v_2,\ldots,v_k$, then we denote ω_f as the weight of v_0, and denote the weight of edges adjacent v_0 by $\omega_d=(\omega_{d1},\omega_{d2},\ldots,\omega_{dk})$. Based on the definition of weight by ω_f and ω_d, we obtained the graph structure of M with vertex and edge weights: ω_f, v_1, ω_{d1}, v_2, $\omega_{d2},\ldots,v_k$, ω_{dk}. Then partition the weighted mesh M into several patches using k-way multilevel constraint segmentation algorithm, the initial boundary is obtained.

Chess Horse Octopus

Fig.2 Pre-Segmentation results

For the mesh with regular size face distribution, the pre-segmentation can give reasonably better results (as in Fig.2). Although the unsupervised k-way partition has a higher performance, however, it suffers from jaggy boundary problem due to noise and other variations in local properties near the cutting boundary, as well as the patches with same size, and even higher precision weight can not always capture salient geometric features (as in Fig.3). Post-Segmentation will address the shortcoming above, to further improve the results, and make the boundary more meaningful.



Original model Pre-Segmentation region Pre-Segmentation border

Fig.3 Pre-Segmentation result of model Teapot

5 Post-Segmentation

In this section, we will address the jaggy boundary and optimize the boundary location of pre-segmentation. The mean curvature is used to detect the feature boundary, and the deformation contour is defined and impeded to move towards the feature boundary by minimizing the boundary strength function. When the function obtains its minimum value, we obtain the final cutting boundary of post-segmentation at the position of the deformation contour.

5.1 Feature boundary detecting

Using the curvature tensor of the mesh surface, G. Lavoué proposed a boundary detection method for CAD model[26]. Andreas Hubeli extracted the boundary feature by computing a collection of piecewise linear curves of mesh salient features, such as edges and ridge lines[27]. Y. Sun proposed an edge detection technique which was based on feature analysis of the surface normal vector field in a geodesic window[28]. Among the above, Hubeli's method can adapt to more different structural feature model and has good robustness. We adopt the idea in Ref. [27], use the mean curvature of vertices on M to detect the shape features, and the selected feature points be defined as the feature boundary.

Firstly, we compute the mean curvature histogram of vertices on M, and then determine the gradient threshold T of the curvature using concave and convex histogram analysis. There are two kind distributions of the vertex mean curvature histogram, which is bimodality and unimodality. For the histogram with bimodal distribution, we take the trough between two peaks as the gradient threshold value T. As an example of model Chess, according to its mean curvature histogram, we have found T=−0.1 works well to classify vertices, the feature boundary of model Chess as illustrated in Fig.4.

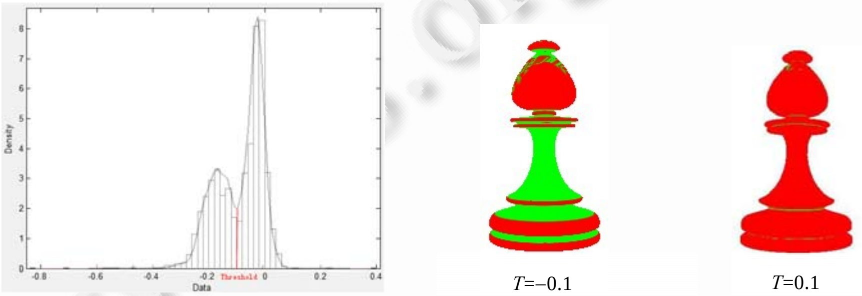


Fig.4 The threshold T of Chess

For the histogram has unimodal distribution, we consider the histogram as a region in the plane, then calculate the convex hull and obtain the maximum convex residuals. The mean curvature corresponding to the maximum convex curvature of the residual value shall be the average gradient threshold T that we desire (as in Fig.5). Convex residual error weighs by concavity estimate[29].

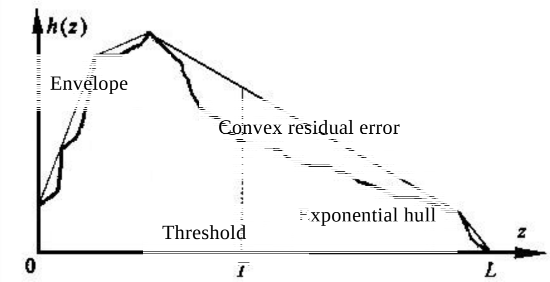
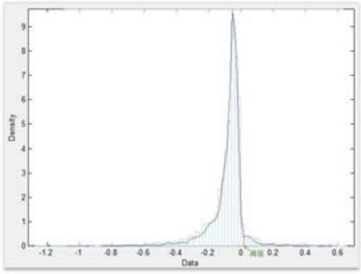


Fig.5 The threshold T under the concave and convex histogram analysis

Exponential hull. We classify the vertices of M with T, and denote the feature boundary as the vertices set whose mean curvature value is greater or equal to threshold T. Take model Teapot as example, we set the threshold T=0.02, the vertices with its mean curvature < T is rendered in red, otherwise it is rendered in green as the feature boundary (as in Fig.6).



Characteristic boundary obtained by the optimal threshold

Fig.6 Threshold T and feature boundary of Teapot

5.2 Post-Segmentation by minimizing the boundary strength function

After pre-segmentation progress, each patch has a closed initial boundary denoted as $\boldsymbol{I}$. In order to adjust $\boldsymbol{I}$ to a more meaningful position, this section defines the function of boundary strength $Edge^*$ on the $\boldsymbol{I}$. When we minimizes $Edge^*$, a deformation contour Φ starting with $\boldsymbol{I}$ is impeded to move toward feature boundary $\boldsymbol{J}$. When $Edge^*$ obtains its minimum value, the movement terminates and the final cutting boundary of post-segmentation resides at the position of the Φ.

Based on Y. Lee's work[30], for a given deformation contour Φ, its boundary strength function is defined as follows:

$$Edge^* = \int_{\Phi} [E_{internal}(v(s)) + E_{external}(v(s))]\mathrm{d}s$$

Where $Einternal(v(s))$ is the internal impetus strength function of Φ, to maintain it's smooth, and $Eexternal\ (v(s))$ is the restraint strength function of J to direct the final position of Φ movement. $v(s)$ is a sequence of sample points on Φ: $v(i)=(x(i), y(i), z(i))$, $i=1,2,\ldots,n$.

We define the internal propelling force function as:

$$Einternal\ v(s) = \frac{\sum_{i=1}^{Mv} \mathrm{H}_i \times \| v(i) - \bar{v}(i) \|^2}{\mathrm{M}v}$$

where Mv is the number of vertices on the deformation contour Φ, H_i is the mean curvature, and $\bar{v}(i)$ is the mean value of $v(i)$. Similarly we define the restraint strength function as:

$$Eexternal\ v(s) = -\frac{\sum_{j=1}^{Nv} \mathrm{T}_j \times \| v(j) - \bar{v}(j) \|^2}{\mathrm{N}v}$$

where T_j is the mean curvature of the vertices on $\boldsymbol{J}$, and $v(j)$ is a sequence of sample points on $\boldsymbol{J}$: $v(j)=(x(j), y(j), z(j))$, $j=1,2,\ldots,n$, $\bar{v}(j)$ is the mean value of $v(j)$, Nv is the number of vertices on $\boldsymbol{J}$.

In implementation, first select a pre-segmentation area arbitrarily, and construct its boundary strength function $Edge^*$. By minimizing the $Edge^*$, the deformation contour Φ is driven to move toward corresponding feature boundary. When $Edge^*$ has its local minimums, this deformation contour resides it finally position, and the segmentation is done with a meaningful result.

6 Experimental Results

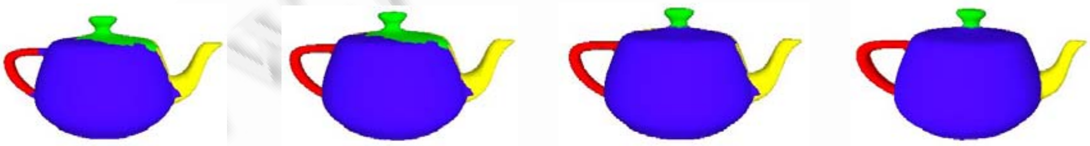
Our method is implemented in Microsoft Visual Studio C++, and run on Intel(R) Core2 processor 2Ghz with 2G RAM, with Shape Benchmark database of Princeton university. The experimental models and segmentation data are shown in Table 1. In the table, only the partitioning time basing on the boundary strength of post-segmentation is shown, as the pre-segmentation is implemented on the k-way multilevel segmentation.

Table 1 Experimental result

Num	Mesh	Number of triangular patches	Patches of segmentation	Time (s)
A	Chess	13 450	7	0.07 7
B	Horse	39 698	6	0.18 9
C	Octopus	14 498	9	0.22 3
D	Mech	31 006	2	0.32 8
E	Teapot	7 800	4	0.15 5
F	Table	27 848	4	0.21 8
G	Vase	13 368	3	0.09 3
H	Hand	3 026	7	0.06 3
I	Giraffe	18 474	8	0.17 1

Consider that two adjacent patches have common boundary, for the patch which has only one neighboring region, the segmentation boundary is unique and the post-segmentation is easy, we can implement the post-segmentation first on the patches which has only one adjacent region. Then the patch has several adjacent regions will obtain its corresponding boundary after the post-segmentation of its adjacent area finished.

In view of different models, their region neighboring situation is different after pre-segmentation. Therefore in the post-segmentation process, different patches need different times to adjust the boundary. Take the Teapot as an example, the pre-segmentation produces four patch regions labeled with different colors (as in Fig.7). In the post-segmentation process, we first adjust the boundary of red zone, to make its initial boundary move towards feature boundary, so part of the blue zone is divided into the red zone. Then make another adjustment on the green area secondly, during this adjustment process, green patches under the boundary movement are divided into the blue area. The third area boundary movement is on the yellow area to get the end of segmentation results. Three times is needed to achieve the final segmentation results of Teapot.



Pre-Segmentation result 1st boundary movement 2nd boundary movement 3rd boundary movement

Fig.7 Steps of post-segmentation process of Teapot

The experiment results show that our method works well, some segmentation results are even better than K-Means method, especially to the model with prominent local marginal, such as Mech, Teapot，Table et al. For those models which have not prominent shape features on its boundary, our algorithm can also get meaningful segmentation results meeting the minimum rules, such as Octopus, Hand, Giraffe (as in Fig.8).

And the experimental time can be close to real-time even for large scale models with more than 10^4 patches. Let n be the number of faces in the mesh M, then the time computational complexity of pre-segmentation is bounded by $O(n)$; the feature boundary detecting takes $O(n)$ too; the post-segmentation costs $O(n^2)$; thus overall time complexity is $O(n+n+n^2)=O(n^2)$. And the time complexity of K-Means method is $O(n^2)$. The experiment results indicate that, most segmentation results of our method surpass the K-Means method obviously (as in Fig.8). More segmentations comparing with Fitting Primitives and Random Walks are shown as in Fig.9.

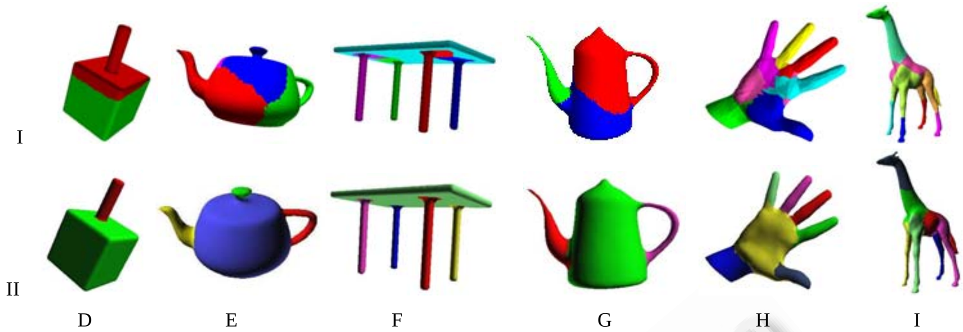


Fig.8 Comparison (Group I: *K*-Means method, Group II: our algorithm)

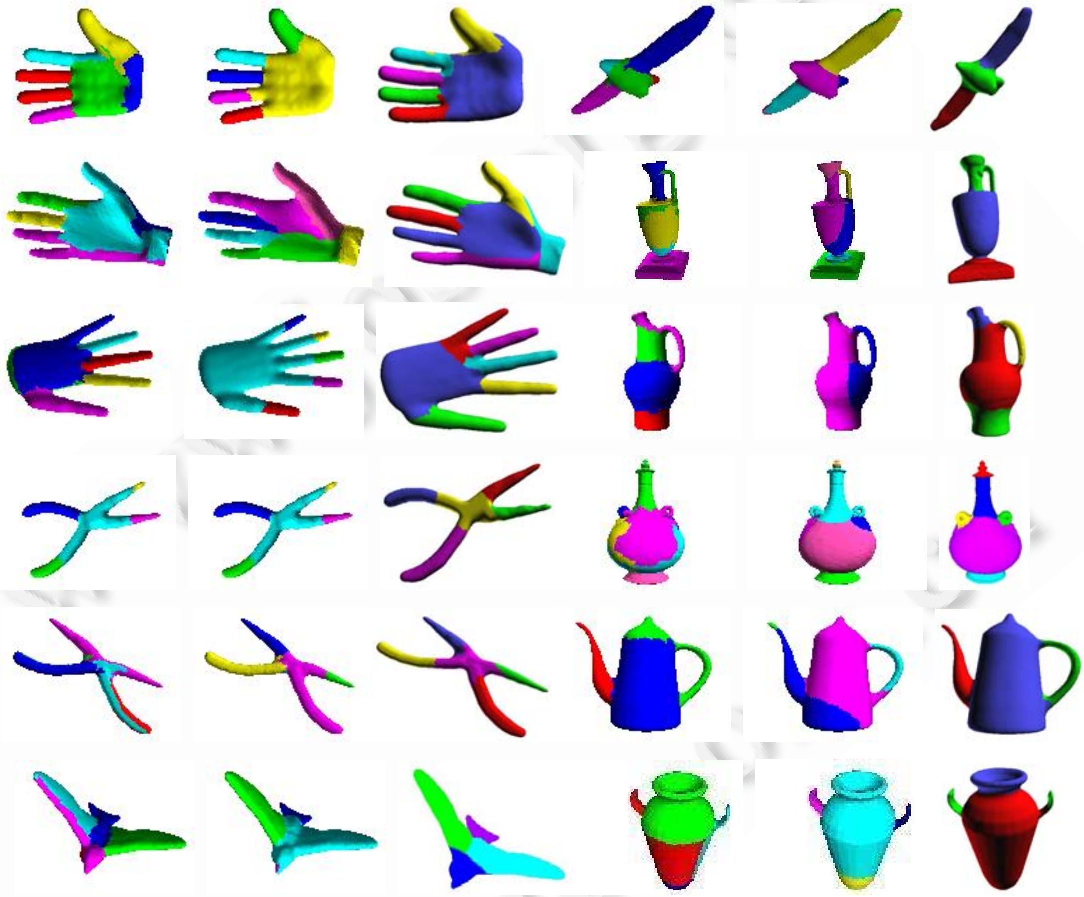


Fig.9 Comparison with fitting primitives and random walks

7 Conclusion

We have presented a novel algorithm for segmentation of meshes, by combining the region feature and the boundary technology, and unifying the k-way division method with the boundary feature detecting. We give a new and effective boundary strength energy function to construct the boundary deformation model, to reduce over segmentation and avoid jagged boundary. Our method has a better adaptability, and is robust for those meshes with well-proportioned distribution or with clear body and branches. For the partial boundary of mesh with prominent shape features, our method has high speed and the robust division results specially.

But the drawback is that our algorithm fails to deal with the mesh with no prominent shape feature, or with a no-closed feature boundary. Some pro-processing must be done to get a closed boundary fist, to improve its robustness.

References:

[1] Funkhouser T, Kazhdan M, Shilane, *et al*. Modeling by example. ACM Trans. On Graph, 2004,23(3):652−663.

[2] Ckler ZO, Stalling M, Hege D, *et al*. Fast and intuitive generation of geometric shape transitions. The Visual Computer, 2000, 16(5):241−253.

[3] Li X, Woon TW, *et al*. Decomposing polygon meshes for interactive applications. In: Proc. of the ACM Symp. on Interactive 3D Graphics. 2001. 35−42.

[4] Katz S, Tal A. Hierarchical mesh decomposition using fuzzy clustering and cuts. ACM Trans. On Graph (SIGGRAPH), 2003,22(3): 954−961.

[5] Varady T, Martin RR, Coxt J. Reverse engineering of geometric models: an introduction. Computer Aided Design, 1997,29(4): 255−268.

[6] Sun XP, Li H. A survey of 3D mesh model segmentation and application. Journal of Computer Aided Design and Computer Graphics, 2005,17(8):1647−1655 (in Chinese with English abstract).

[7] Mangan AP, Whitaker RT. Partitioning 3d surface meshes using watershed segmentation. IEEE Trans. on Visualization and Computer Graphics, 1999,5(4):308−321.

[8] Katz S, Leifman G, Tal A. Mesh segmentation using feature point and core extraction. The Visual Computer, 2005,21(8):865−875.

[9] Gelfand N, Guibas LJ. Shape segmentation using local slippage analysis. In: Proc. of the Eurographics Symp. on Geometry Processing. 2004. 219−228.

[10] Lai YK, Hu SM, Martin RR, Rosin PL. Rapid and effective segmentation of 3D models using random walks. Computer Aided Geometric Design, 2009,26(6):665−679.

[11] Fan J, Yau DY, Elmagarmid AK, Aref WG. Automatic image segmentation by integrating color-edge extraction and seeded region growing. IEEE Trans. on Image Process, 2001,10(10):1454−1466.

[12] Aslan C. Disconnected skeletons for shape recognition [MS. Thesis]. Department of Computer Engineering, Middle East Technical University, 2005.

[13] Tari S, Shah J, Pien H. Extraction of shape skeletons from grayscale images. CVIU, 1997,66(2):133−146.

[14] Mumford D, Shah J. Optimal approximations by piecewise smooth functions and associated variational problems. Commun. Pure Appl. Math., 1989,42(5):577−685.

[15] Ambrosio L, Tortorelli V. On the approximation of functionals depending on jumps by elliptic functionals via Γ-convergence. Commun. Pure Appl. Math., 1990,43(8):999−1036.

[16] Erdem E, Erdem A, Tari S. Edge strength functions as shape priors in image segmentation. In: Proc. of the EMMCVPR. 2005. 490−502.

[17] Erdem E, Tari S, Vese L. Segmentation using the edge strength function as a shape prior within a local deformation model. In: Proc. of the 16th IEEE Int'l Conf. on Image Processing (ICIP 2009). Cairo, 2009. 2989−2992.

[18] Chan T, Vese L. Active contours without edges. IEEE Trans. on Image Processing, 2001,10(2):266−277.

[19] Shi JB, Malik J. Normalized cuts and image segmentation. IEEE Trans. on PAMI, 2000,22(8):888−905.

[20] Shlafman S, Tal A, Katz S. Metamorphosis of polyhedral surfaces using decomposition. Computer Graphics Forum, 2002,21(3): 219−229.

[21] Dyn N, Hormann K, Kim SJ, Levin D. Optimizing 3D triangulations using discrete curvature analysis. In: Mathematical Methods for Curves and Surfaces. 2000. 135−146.

[22] Fang HL, Wang GJ. Comparison and analysis of discrete curvatures estimation methods for triangular meshes. Journal of Computer Aided Design and Computer Graphics, 2005,11(17):2501−2506.

[23] Abou-Rjeili A, Karypis G. Multilevel algorithms for partitioning power-law graphs. In: Proc. of the IEEE Int'l Parallel & Distributed Processing Symp. 2006.

[24] Karypis G, Kumar V. A fast and highly quality multilevel scheme for partitioning irregular graphs. SIAM Journal on Scientific Computing, 1998.

[25] Karypis G, Kumar V. Multilevel *k*-way partitioning scheme for irregular graphs. Journal of Parallel and Distributed Computing, 1998,48(1):96−129.

[26] Lavoue G, Dupont F, Baskurt A. A new CAD mesh segmentation method based on curvature tensor analysis. Computer Aided Design, 2005,37(10):975−987.

[27] Hubeli A, Meyer K, Gross M. Mesh edge detection. Technical Report, 351, ETH Zürich, Institute of Computer Systems, 2000.

[28] Sun Y, Page DL, Paik JK, Koschan A, Abidi MA. Triangle mesh-based edge detection and its application to surface segmentation and adaptive surface smoothing. In: Proc. of the Int'l Conf. on Image Processing ICIP 2002, Vol.III. 2002. 825−828.

[29] Zhang YJ. Image Segmentation. Beijing: Science Press, 2001. 34−35 (in Chinese).

[30] Lee Y, Lee S. Geometric snakes for triangular meshes. Computer Graphics Forum, 2002,21(3):229−238.

附中文参考文献:

[6] 孙晓鹏,李华.三维网格模型分割及应用综述.计算机辅助设计与图形学学报,2005,17(8):1647−1655.

[29] 章毓晋.图像分割.北京:科学出版社,2001.34−35.

SUN Xiao-Peng was born in P. R. China and received a PhD in Computer Science from Institute of Computing Technology, Chinese Academy of Sciences. He is a distinguished professor of LNNU, and his research interests include graphics and VR.

JI Yan-Jie was born in P. R. China, and her research interests include graphics and VR.

WEI Xiao-Peng was born in P. R. China and received a PhD in Mechanical Engineering from Dalian University of Technology. He is a professor at DLUT, and his research interests include graphics.