

# CEFM

## *A Heuristic Mesh Segmentation Method based on Convexity Estimation and Fast Marching*

Jun Zhang<sup>1,2</sup>, Zhouhui Lian<sup>1</sup>, Zhenbao Liu<sup>2</sup> and Jianguo Xiao<sup>1</sup>

<sup>1</sup>*Institute of Computer Science and Technology, Peking University, Beijing, P.R.China*

<sup>2</sup>*School of Aeronautics, Northwestern Polytechnical University, Xi'an, P.R.China*

Keywords: Mesh Segmentation, Shape Descriptor, Convexity, Fast Marching, Local Depth.

Abstract: Mesh segmentation is a fundamental way of shape analysis and understanding for 3D mesh models. In this paper, we propose an effective heuristic mesh segmentation algorithm, which is based on concave areas detection and heuristic 2-category classification via fast marching. The algorithm has several merits. First, the boundary between each pair of segments is close to the natural seams of 3D objects. Second, it is robust against pose variations and isometric transformations. Finally, our algorithm decomposes non-rigid 3D models into a set of rigid components in a short period of time and the procedure is fully automatic. Extensive experiments in this paper demonstrate that the proposed method outperforms the state of the art in mesh segmentation.

## 1 INTRODUCTION

3D mesh models are now widely available and used in various applications due to the rapid development of 3D acquisition techniques. As a result, the demand for mesh analysis and understanding is ever increasing. Segmenting 3D models into parts with semantic information is a fundamental problem in geometric processing. Many relevant tasks such as shape retrieval, skeleton extraction, mesh editing and so on, require mesh segmentation as a preprocessing step. However, until now, how to automatically and properly segment 3D models into parts which are consistent with human perception is still a challenging problem.

In (Shapira et al., 2008), the authors proposed a novel local shape descriptor named shape-diameter function (SDF) and applied it to mesh segmentation. The SDF descriptor describes volumetric information of mesh models by measuring the diameter of local parts, which helps to guide volumetric part extraction and shape segmentation. The major limitation of SDF-guided segmentation is that it is not able to decompose non-rigid parts with similar volumetric information (e.g., in Fig.3(d), two thighs of a human model are not separated), most of which is undesirable for human perception and real applications. Lai *et al.* (Lai et al., 2008) extended the original random walks segmentation method from 2D images to

3D meshes. The main idea of this work is to automatically select a couple of seeds (representing the labels of different parts) from a mesh model to achieve oversegmentation, and then merge the parts based on the similarity of neighboring parts. The method is fast and easy to implement, but it tends to over segment most of the mesh models, which goes against human perception.

Inspired by the aforementioned methods, we design an effective automatic segmentation algorithm. Specifically, we improve the SDF-guided segmentation method and use it as an initial segmentation process, and then propose a heuristic algorithm to implement further segmentation of subparts. The latter is based on the detection of concave areas and 2-category classification. The basic idea of our heuristic algorithm is to firstly set a criterion to estimate each segmented part obtained after the initial segmentation process and decide whether it should be further segmented or not. If the criterion is satisfied, the part will then be further segmented by a fast-marching-based method (see Section 3.2.3). The algorithm repeats this procedure until there is no change in any segmented part. We have tested our method elaborately with the Princeton Segmentation Benchmark (Chen et al., 2009) and the experimental results indicate that our method outperforms the state-of-the-art methods. The remainder of this paper is organized as follows. Section 2 briefly reviews related work. An effective

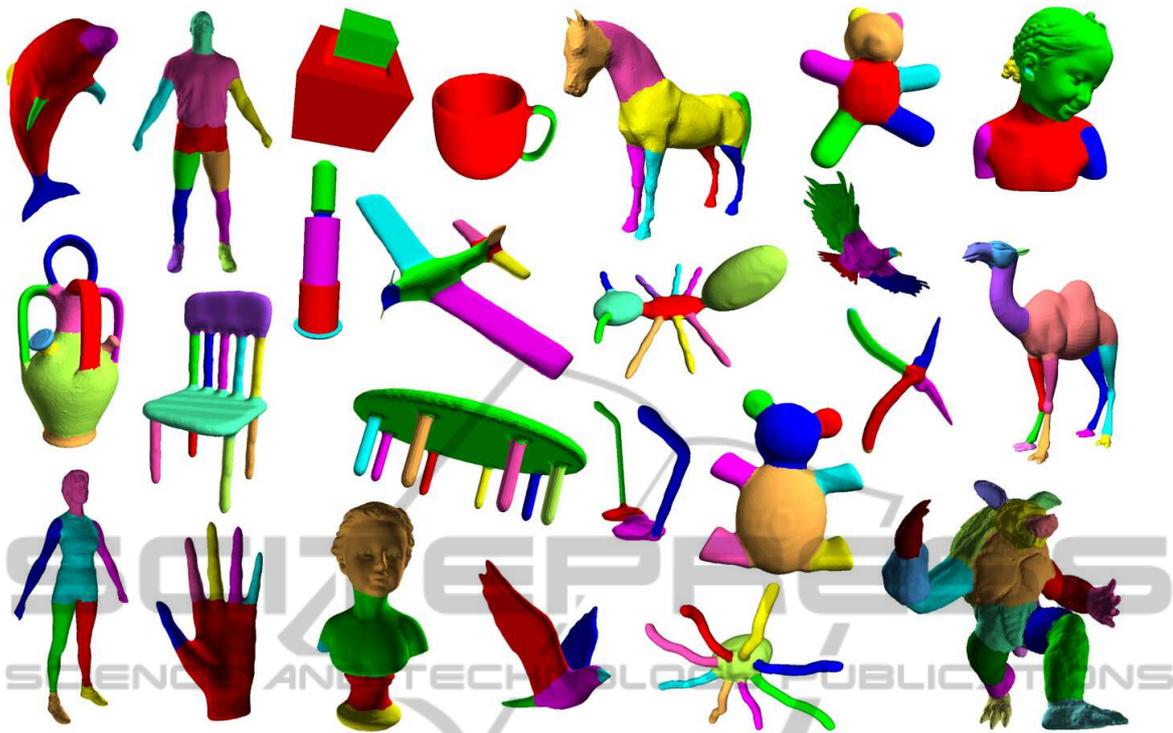


Figure 1: Segmentation results of representative models in the Princeton Segmentation Benchmark (Chen et al., 2009) using our proposed method. It can be clearly seen that our segmentation boundaries naturally follow the seams of shapes.

mesh segmentation algorithm based on concave areas detection and heuristic 2-category classification is presented in Section 3. Experimental results and discussions are given in Section 4, following with conclusions in Section 5.

## 2 RELATED WORK

Generally, existing mesh segmentation methods can be classified into two categories based on the types of models they aim to segment. Algorithms of the first category are designed for applications such as reverse engineering of CAD models. These methods segment models into patches, and each patch is considered as the best fit to one of the given mathematical surfaces, such as planes, cylinders and spheres, etc. The second category consists of algorithms that concentrate on segmenting natural shapes into semantic components which correspond well with human intuition. Our algorithm mainly aims at solving the latter problems.

Numerous approaches for 3D shapes segmentation have been explored in the last few years. For instance, Shlafman *et al.* (Shlafman et al., 2002) used k-means clustering to segment the models into meaningful parts. Katz *et al.* (Katz et al., 2003)

improved the above-mentioned approach by using fuzzy clustering and graph cut methods to achieve smoother boundaries between clusters. These methods are mainly based on iterative clustering, which can be easily achieved. But the necessity to compute pairwise distances makes them computationally expensive to handle high-resolution models directly. To solve this problem, Zhang and Liu (Zhang and Liu, 2005), (Liu and Zhang, 2007) employed spectral clustering methods to analyze meshes in 2D space. While Lai *et al.* (Lai et al., 2008) extended random walks method originally used for image segmentation to the mesh segmentation problem. A common feature of the methods mentioned above is that they all include concavity information in their underlying algorithm, which ensures that the boundaries between segmented parts adhere to concave areas. However, all these methods use a single set of seeds or cluster centers to guide the segmentation procedure. As a result, they rely heavily on the number of seeds or clusters.

There also exist some other more complex methods for mesh segmentation. For example, Katz *et al.* (Katz et al., 2005) proposed a method based on core extraction using centrality information to determine the prominent parts of the models. The randomized cuts method was proposed in (Golovinskiy and Funkhouser, 2008) to generate a random set of

mesh segmentations and then measure how often each edge of the mesh lies on a segmentation boundary in the randomized set to accomplish final segmentation. Kalogerakis *et al.* (Kalogerakis et al., 2010) trained a conditional random field probabilistic model for simultaneous segmentation and labeling of parts in 3D meshes. They fused hundreds of informative features such as curvature, shape context, etc, in the training procedure to achieve good segmentation performance. Nevertheless, most of these alternative methods involve complicated procedures, which result in high computation cost.

Our method employs a modified SDF-guided segmentation method as an initial segmentation process, which is fast and effective for coarse partition of mesh models. After that, a heuristic algorithm based on convexity estimation and fast marching methods (Sethian, 1999) is utilized to implement further subparts segmentation. Compared to methods mentioned above, the method proposed in this paper is robust and effective with less parameters and low computation cost.

### 3 METHOD DESCRIPTION

#### 3.1 Initial Segmentation

As mentioned in Section 1, the first step of our method is to implement an initial segmentation procedure. The aim of this procedure is to decompose any mesh model into a principal part and other branch parts in a coarse way. One reason why we do this initial segmentation is to avoid over-segmentation and bad segment boundaries, as the heuristic strategy is difficult to control when the mesh model is complex in topological structure. Another reason is that, with the initial segmentation process, the time costs of the whole algorithm can be markedly reduced. In other words, the initial segmentation step takes charge of the general decomposition while the heuristic segmentation step executes the detailed work.

When segmenting 3D meshes, we observe that the subparts of a given model to be separated contain certain volumetric information, which is distinguishable among different subparts. Therefore, a proper solution is needed to capture these volumetric information for segmentation. We employ an efficient segmentation procedure by utilizing *shape diameter function* (SDF) (Shapira et al., 2008) descriptor (see Fig.2). The SDF descriptor connects volumetric information of a mesh model onto its boundary mesh by measuring the local diameter of the model at faces on its

boundary. Hence, it is suitable to guide volumetric part extraction and shape segmentation.

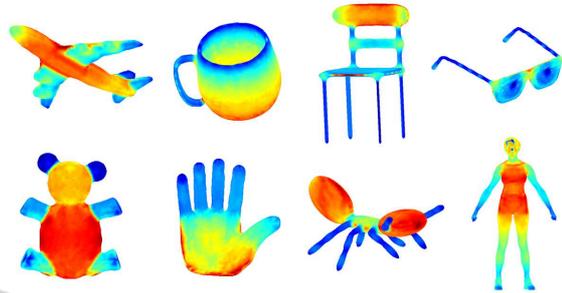


Figure 2: Some typical models from the Princeton Segmentation Benchmark model set. The color mapping on the models is the normalized SDF values. As we can see that these values are able to indicate similarity between analogous parts.

Specifically, our method first builds a Gaussian Mixture Model (GMM) with the Expectation-Maximization (EM) (Bilmes, 1997) algorithm to fit  $k$  Gaussian functions to the histogram of SDF values of faces on the mesh. By using GMM, we can compute a vector  $v_f \in R^k$  for each face  $f$ , where  $v_f^i$  is the probability of  $f$  belonging to the  $i$ th Gaussian function. Here,  $k$  is experimentally chosen as 3 for most models. For some models with simple topological structure, the value is set to  $k = 2$ . If simply choosing the most probable index of Gaussian function to label each  $f$ , we can obtain an initial result as demonstrated in Fig.3(b). This is undesirable because it doesn't include contextual information and thus results in large numbers of holes and jags.

To improve the initial segmentation results obtained above, we take contextual information of the mesh into account and make sure that the boundaries between segmented parts are smooth and approach local mesh features such as concave areas or creases. We employ an alpha expansion graph-cut algorithm (Boykov et al., 2001) to fulfil the goal. The graph cut procedure assigns a single segment label to each face, considering both the probability vector from the EM step and the contextual information. Specifically, The graph cut procedure minimizes the following energy function:

$$E(\hat{x}) = \sum_{f \in F} e_1(f, \hat{x}(f)) + \lambda \sum_{(f,g) \in E} e_2(\hat{x}, f, g) + \mu \sum_{(f,g) \in E} e_3(\hat{x}, f, g), \quad (1)$$

$$e_1(f, \hat{x}(f)) = -\log(P(f|\hat{x}(f)) + \epsilon), \quad (2)$$

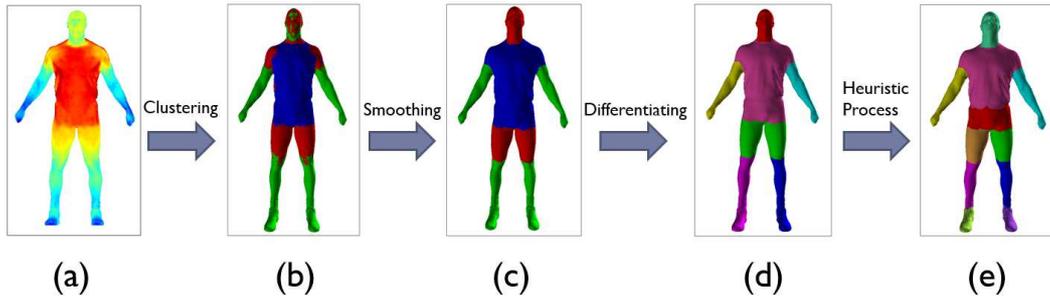


Figure 3: Process of our segmentation method. (a) Distribution of the SDF values. (b) Result of clustering only. (c) Modified segmentation with contextual information. (d) Differentiating the similar parts. (e) The final segmentation result.

$$e_2(\hat{x}, f, g) = \begin{cases} l_{(f,g)}(1 - \log(\theta_{(f,g)}/\pi)) & \hat{x}(f) \neq \hat{x}(g), \\ 0 & \hat{x}(f) = \hat{x}(g), \end{cases} \quad (3)$$

$$e_3(\hat{x}, f, g) = \begin{cases} l_{(f,g)}(1 - \log(d_{(f,g)} + \epsilon)) & \hat{x}(f) \neq \hat{x}(g), \\ 0 & \hat{x}(f) = \hat{x}(g), \end{cases} \quad (4)$$

where

- $\hat{x}$  denotes the face label.
- $P(f|\hat{x}(f))$  refers to the probability of assigning face  $f$  to label  $\hat{x}(f)$ . These values are derived from the normalized summation values.
- $\theta_{(f,g)}$  represents the dihedral angle between any connected pair of faces  $f$  and  $g$ .
- $d_{(f,g)}$  represents the difference between SDF values of any connected pair of faces  $f$  and  $g$ .
- $l_{(f,g)}$  is the length of the edge shared by  $f$  and  $g$ .
- $F$  and  $E$  are the face set and edge set respectively.
- $\lambda$ ,  $\mu$  and  $\epsilon$  are constants. Experimentally, we set  $\lambda$  to 1,  $\mu$  to 0.8 and  $\epsilon$  to  $10^{-3}$ .

We consider  $e_1$  as a *data term*,  $e_2$  and  $e_3$  as *smoothness terms*. Note that the main difference between our method and the original one proposed in (Shapira et al., 2010) is the introduction of  $e_3$ , which is of great significance in improving the segmentation results.  $e_2$  helps to contract the boundaries between parts to the concave areas. However, if we put more emphasis on  $e_2$  (by increasing the value of  $\lambda$ ), most of the correctly-segmented parts will merge together. On the other hand, the segmentation result won't be smooth and will attach with many undesired little patches if we assign  $\lambda$  with a small value (see Fig.4(a)). We introduce  $e_3$  to play a role as a trade-off to fix this problem. The  $e_3$  term measures how similar each pair of adjacent faces is based on the dissimilarity between their features. Under the impact

of  $e_3$ , labels of any adjacent faces will stay the same if the feature distance between them is small enough. Otherwise, they will be different. The effectiveness of  $e_3$  is demonstrated in Fig.4(b). As shown in Fig.9, our further experiments verify that the  $e_3$  term contributes greatly to the final segmentation results.

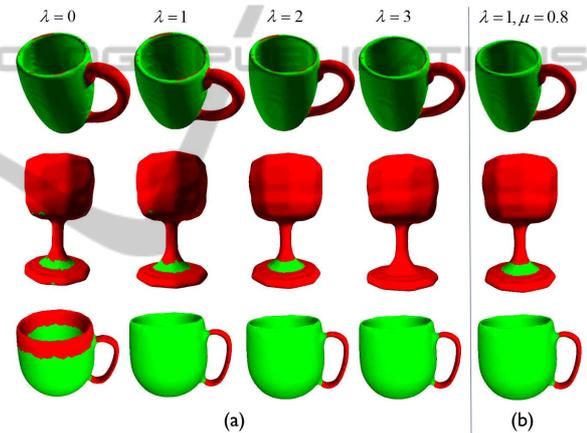


Figure 4: Our initial segmentation results of some cup models over a range of parameters. (a) The parameter  $\lambda$  of smooth term  $e_2$  ranging from 0 to 3. We can see that the values of  $\lambda$  suitable for various models are not always the same. (b) Combined action of smooth term  $e_2$  and  $e_3$ . With the help of  $e_3$ , the initial segmentation results become smoother and more robust.

After above-mentioned procedures, we observe that similar parts are still assigned with the same labels, see Fig.3(c). We obtain the actual parts via propagating from a seed face to differentiate the parts which are not connected but with the same label. Up to now, the final results of initial segmentation process have been acquired (see Fig.3(d)).

## 3.2 Heuristic Segmentation

As previously mentioned, the goal of the initial segmentation process is to coarsely decompose any mesh model into a principal part and other branch parts.

For example, a human model will be partitioned into torso, head, arms and legs after the initial processing. However, it is far from being enough to achieve a better and natural segmentation result and reach the goal to decompose a non-rigid 3D model into a set of rigid components. Thereby, we design a heuristic segmentation procedure to execute further decomposition for those initially segmented parts.

### 3.2.1 Criterion for Part-segment

Each part in a certain model obtained after the initial segmentation process should be estimated whether to be segmented again or not. Our strategy is to calculate the convexity value of each part and set a threshold. If the convexity value is under the threshold, then this part is suggested to be segmented again. Otherwise it remains the same. Numerous experimental results demonstrate that the most appropriate threshold value can be selected as 0.85. The convexity value  $C$  of a part  $P$  is defined as follows,

$$C(P) = \frac{\text{Volume}(P)}{\text{Volume}(CH(P))}, \quad (5)$$

where  $CH(P)$  is the convex hull of part  $P$ . As illustrated in Fig.5 which shows the volume of each part and the corresponding convex hull of a human model, we can easily decide which parts should be segmented again (i.e., crus, thigh and torso maybe) or not (i.e., head and hands). Therefore, the criterion defined above is able to make sure that all non-rigid parts can be segmented again.

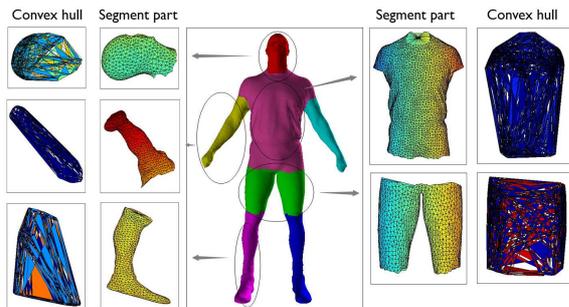


Figure 5: Overview of segment parts of a human model. In the center is an initial segmentation result of the human model, and on both sides are segmented parts and their corresponding convex hulls.

### 3.2.2 Local Depth

The local depth (Lin et al., 2010) feature measures how deep a position is located with respect to its neighbors. The basic idea of local depth comes from the observation of Fig.6(b). As we can see, there is

an obvious height difference between  $D$  and  $E$ , but the height difference between  $D$  and  $A$  is more suitable to measure the location of point  $D$  according to human perception. One effective way of presenting this feature is to calculate its local depth.

There are four steps to compute the local depth of a given vertex  $v$  on a mesh  $M$ . For each vertex on a mesh, we first determine its height direction on its normal direction. The normal of a vertex is a weighted sum of normals of its adjacent faces. Suppose  $F_v = \{f \mid f \text{ is the face adjacent to } v\}$ , then the normal of  $v$  can be estimated as

$$\vec{n}_v = \frac{\sum_{f_i \in F_v} \vec{n}(f_i) \cdot \text{Area}(f_i)}{\sum_{f_i \in F_v} \text{Area}(f_i)}, \quad (6)$$

Then we define the local neighbour. In Fig.6(b), if we only consider one ring neighborhood, the depth of point  $D$  is the difference between the heights of  $E$  and  $D$ . But if we consider a wider range of neighborhood, such as three rings, then the depth of point  $D$  becomes the difference between  $A$  and  $D$ , which is more reasonable. we set the ring number value to 3 in all the experiments and obtain satisfactory results. Thirdly, for each vertex  $v$ , we compute the height differences between  $v$  and its neighbors to form the set  $D_v = \{d_{uv} \mid d_{uv} \text{ is the height difference between } v \text{ and } u\}$ . The height difference  $d_{uv}$  is computed using Eq.7. Finally, We select the largest value from  $D_v$  as the local depth of  $v$ .

$$d_{uv} = \cos \alpha \cdot |\vec{u}\vec{v}| = \frac{\vec{n}_u \cdot \vec{u}\vec{v}}{|\vec{n}_u| \cdot |\vec{u}\vec{v}|} \cdot |\vec{u}\vec{v}| = \vec{n}_u \cdot \vec{u}\vec{v}, \quad (7)$$

From the definition of local depth, we can observe that, firstly, a positive local depth indicates that the vertex is at the concave position, and a negative one indicates convex position. Secondly, the larger the positive local depth of a vertex is, the more possible it is for the vertex to be a boundary point, as the surface is more bent, see Fig.6(a). The local depth feature has been proved to consider a wider range of geometry information than the curvature feature does via large amounts of experiments conducted in (Lin et al., 2010).

### 3.2.3 Fast-marching Based Segmentation

The ultimate goal of our method is to decompose any non-rigid 3D model into a set of rigid components, meanwhile ensuring that the boundary between each pair of segmented parts should be very close to the natural seams of the model. To achieve this goal, we implement a further segmentation processing if a certain part, which comes from the initial segmentation

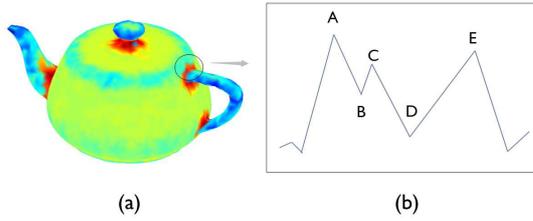


Figure 6: (a) The distribution of local depth values of a teapot model. Vertices in the concave place of the mesh have large local depth values, which are in red color. (b) Schematic diagram of local depth.

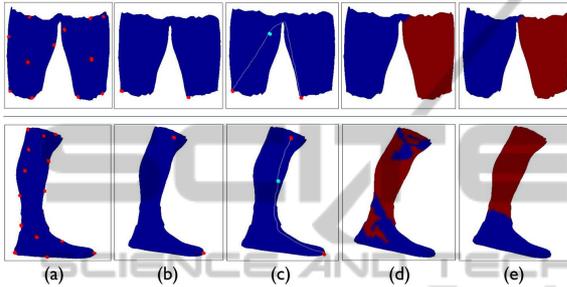


Figure 7: Demonstration of our heuristic segmentation. (a) Points sampling. (b) Seed points selection. (c) Pathways of an arbitrary point towards both the seeds. (d) Labeling according to the pathway local depth summation. (e) Smoothing the labeling result.

process, satisfies the convexity estimation criterion. There are mainly five steps in this procedure.

*Uniform sampling.* In this step, we need to compute the geodesic distance between certain pairs of vertices. As most of the mesh models we deal with contain tens of thousands of vertices and faces, it's rather time-consuming and impractical to compute the geodesic distance between any pair of vertices in a model. As a result, we perform a uniform sampling to obtain a certain number of vertices which are evenly distributed in the segment part, as demonstrated in Fig.7(a). we set the number of sample points to 20 in this paper.

*Selecting seeds.* Now we calculate both the geodesic distance (based on fast marching methods (Sethian, 1999)) and the Euclidean distance between each pair of vertices in the sample point set. And then, for each two sampled vertices, we calculate the ratio between the geodesic distance and the Euclidean distance as follows,

$$r_{uv} = \frac{GD_{uv}}{ED_{uv}}, \quad (8)$$

where  $r_{uv}$  is the distance ratio of vertices pair  $u$  and  $v$ ,  $GD$  denotes the geodesic distance and  $ED$  the Euclidean distance. Then we sort the geodesic dis-

tance values calculated above in descending order, and from the first ten pairs we choose a pair of vertices as seeds whose distance ratio is the largest (see Fig.7(b)). In most situations, the geodesic pathway of this pair is able to cross the most bent area in the segment part. We don't simply choose the pair of vertices with largest geodesic distance, because in certain cases (e.g., Fig.8(c)) such processing results in incorrect segmentations while our results (e.g., Fig.8(d)) are satisfactory.

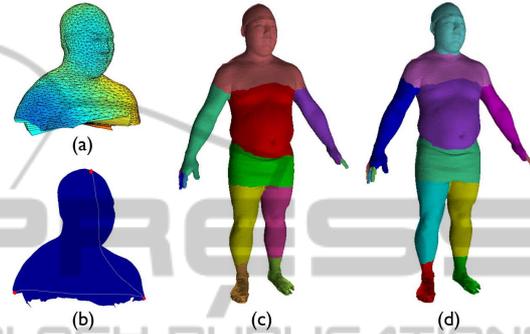


Figure 8: A special situation of a man model segmentation procedure during seeds selection. (a) The upper body part which is difficult to pick up the seeds. (b) Demonstration of proper seeds selection. The pairs in both arms share the largest geodesic distance, but the other pairs with larger distance ratio are supposed to be the seeds. (c) Segmentation result without taking distance ratio into consideration. (d) Segmentation result that is in consideration of distance ratio.

*Computing the Local Depth Summation.* We compute the geodesic pathway of every vertex to the seeds using fast marching methods (Sethian, 1999). Then for each vertex (except for seeds), we add up the local depth values in each pathway to the seeds (see Fig.7(c)).

*Labeling the Part.* We assign the two seeds with different labels (e.g., 0 and 1). For each vertex, we compare the summation values of each pathway. If the pathway of a certain vertex to one of the seeds has a smaller summation value, the label of this vertex should be consistent with that of the seed, as shown in Fig.7(d).

*Smoothing the Result.* As we can see from Fig.7(d), in most cases, the labeling results do not satisfy our requirements that the boundaries between parts are smooth and adhere to concave areas or creases. To address the problem, we employ the alpha expansion graph-cut algorithm (Boykov et al., 2001) again to smooth the labels. Different from the initial segmentation process, here we just use  $e_1$  and  $e_3$  to construct the smooth term in the energy function. Fig.7(e) shows the final segmentation result after this step.

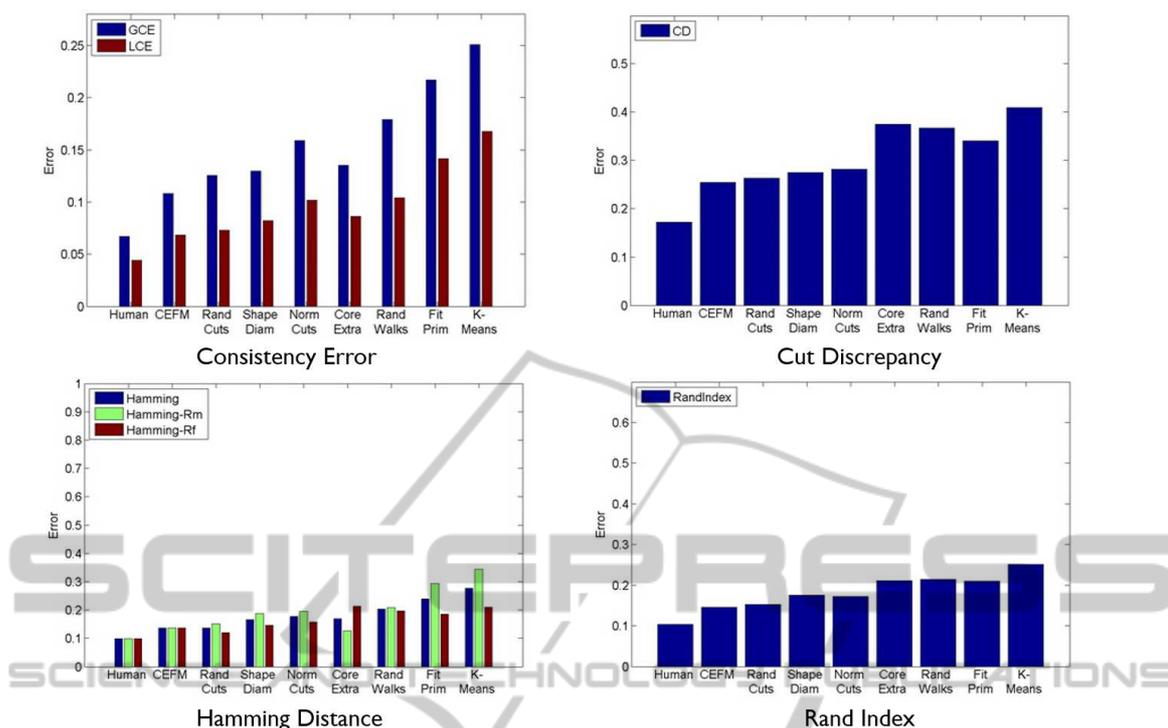


Figure 9: Evaluations using the performance measures proposed in (Chen et al., 2009) and the manual segmentations in the Princeton Segmentation Benchmark. Our approach (short for 'CEFM') clearly outperforms existing state-of-the-art automatic segmentation approaches.

Until now, we have obtained a final segmentation result, as shown in Fig.3(e). Note that the whole segmentation procedure is completely automatic.

## 4 RESULTS

We have applied our method to segment a wide range of models, including the whole set of models in the Princeton Segmentation Benchmark (Chen et al., 2009). As shown in our experiments, for most of the models, the proposed approach is able to generate boundary cuts that adhere to the natural seams of the models.

All 19 categories of 380 models in Princeton Segmentation Benchmark were tested by our method. As we can see from Fig.1, most segmentation results are satisfactory, especially for those models with obvious salient parts, such as ants, bears, airplane, and etc. We evaluated the segmentation results using the proposed metric methods in (Chen et al., 2009). Fig.9 presents the quantitative results of our method and other automatic segmentation methods. Clearly, our easily-implemented and efficient method outperforms existing state-of-the-art methods, including complex methods like randomized cuts (Golovin-

skiy and Funkhouser, 2008). As shown in the bottom left of Fig.9, the Hamming Distance of *missing rate* ( $R_m$ ) of our method is lower while *false alarm rate* ( $R_f$ ) is higher compared with randomized cuts method. This is because our approach tends to segment shapes into more parts due to the heuristic strategy. In terms of computation complexity, our approach is quite efficient and feasible. For a typical model consisting of 10K triangles, the time needed for a total computation process is less than 15 seconds on an Intel 2.3GHz laptop with 4GB memory (codes were written in Matlab under Windows 7).

However, there also exist several drawbacks in our approach. Firstly, our method is not able to deal with shapes containing inadequate or too similar volumetric information, see Fig.10(a). This is because the SDF descriptor has difficulty in distinguishing separated parts of these shapes when detecting diameters inside the shapes. Another drawback is that the results tend to be over-segmented if the shapes contain a lot of folds on their surfaces (Fig.10(b)). Finally, for certain models, the boundaries between segment parts are not smooth enough and look unnatural, as shown in Fig.10(c).

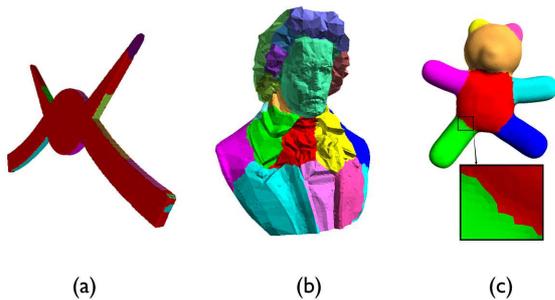


Figure 10: Limitations of our approach. (a) A sheet-like model from SHREC'11(Lian et al., 2011) which is hard to be segmented. (b) A bust model with lots of details. (c) A bear model segmented with coarse boundaries.

## 5 CONCLUSION

In this paper, we introduced an efficient and effective automatic mesh segmentation method based on the detection of concave areas and heuristic 2-category classification via fast marching. We demonstrated the effectiveness of our method by carrying out segmentation experiments on the Princeton segmentation benchmark. Experimental results indicate that our method clearly outperforms existing state-of-the-art approaches in mesh segmentation.

## ACKNOWLEDGEMENT

This work was supported by National Natural Science Foundation of China (Grant No.: 61202230 and 61472015), National Hi-Tech Research and Development Program (863 Program) of China (Grant No.: 2014AA015102).

## REFERENCES

- Bilmes, J. A. (1997). A gentle tutorial on the em algorithm and its application to parameter estimation for gaussian mixture and hidden markov models.
- Boykov, Y., Veksler, O., and Zabih, R. (2001). Fast approximate energy minimization via graph cuts. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 23(11):1222–1239.
- Chen, X., Golovinskiy, A., and Funkhouser, T. (2009). A benchmark for 3D mesh segmentation. *ACM Transactions on Graphics*, 28(3):Article No. 73.
- Golovinskiy, A. and Funkhouser, T. (2008). Randomized cuts for 3D mesh analysis. *ACM Transactions on Graphics*, 27(5).
- Kalogerakis, E., Hertzmann, A., and Singh, K. (2010). Learning 3D mesh segmentation and labeling. *ACM Transactions on Graphics*, 29(3):1–11.
- Katz, S., Leifman, G., and Tal, A. (2003). Hierarchical mesh decomposition using fuzzy clustering and cuts. *ACM Trans. Graphics*, 22:954–961.
- Katz, S., Leifman, G., and Tal, A. (2005). Mesh segmentation using feature point and core extraction. *The Visual Computer*, 21(8):649–658.
- Lai, Y., Hu, S., Martin, R. R., and Rosin, P. L. (2008). Fast mesh segmentation using random walks. In *Proceedings of ACM symposium on Solid and Physical Modeling*, pages 183–191.
- Lian, Z., Godil, A., Bustos, B., Daoudi, M., Hermans, J., Kawamura, S., Kurita, Y., Lavou, G., Nguyen, H., Ohbuchi, R., Ohkita, Y., Ohishi, Y., Porikli, F., Reuter, M., Sipiran, I., Smeets, D., Suetens, P., Tabia, H., and Vandermeulen, D. (2011). Shrec11 track: Shape retrieval on non-rigid 3d watertight meshes. *Eurographics Workshop on 3D Object Retrieval*.
- Lin, J., Yang, Y., Lu, T., He, G., and Ruan, J. (2010). Mesh segmentation by local depth. *2010 Second International Conference on Computer Modeling and Simulation*.
- Liu, R. and Zhang, H. (2007). Mesh segmentation via spectral embedding and contour analysis. In *Computer Graphics Forum (Special Issue of Eurographics 2007)*. Blackwell Publishing.
- Sethian, J. (1999). *Level Set Methods and Fast Marching Methods Evolving Interfaces in Computational Geometry, Fluid Mechanics, Computer Vision, and Materials Science*. Cambridge University Press, London, 1st edition.
- Shapira, L., Shalom, S., Shamir, A., Cohen-Or, D., and Zhang, H. (2010). Contextual part analogies in 3d objects. *International Journal of Computer Vision*, pages 309–326.
- Shapira, L., Shamir, A., and Cohen-Or, D. (2008). Consistent mesh partitioning and skeletonisation using the shape diameter function. *The Visual Computer*, 24(4):249–259.
- Shlafman, S., Tal, A., and Katz, S. (2002). Metamorphosis of polyhedral surfaces using decomposition. *Computer Graphics Forum*, 21(3):219–228.
- Zhang, H. and Liu, R. (2005). Mesh segmentation via recursive and visually salient spectral cuts. In *Proc. of Vision, Modeling, and Visualization*. VMV.