Computers & Graphics 51 (2015) 136–145

Contents lists available at ScienceDirect

Computers & Graphics

journal homepage: www.elsevier.com/locate/cag

SMI 2015

Shape segmentation by hierarchical splat clustering

Huijuan Zhang [a], Chong Li [a], Leilei Gao [a], Sheng Li [a,b], Guoping Wang [a,b,c,*]

[a] Graphics and Interaction Lab, School of Electronics Engineering and Computer Science, Peking University, Beijing 100871, China
[b] Beijing Engineering Technology Research Center of Virtual Simulation and Visualization, Peking University, Beijing 100871, China
[c] Cooperative Medianet Innovation Center, Beijing 100871, China

ARTICLE INFO

Article history:
Received 12 March 2015
Received in revised form
15 May 2015
Accepted 16 May 2015
Available online 28 May 2015

Keywords:
Segmentation
Similarity metric
Patch-aware
Part-aware
Hierarchical clustering

ABSTRACT

This paper presents a novel hierarchical shape segmentation method based on splats for 3D shapes. The major contribution is to propose a new similarity metric based on splats, which combines patch-aware similarity and part-aware similarity adaptively. An optimized $L^{2,1}$ metric for VSA (variational shape approximation) method is used to get splats first, and such adaptive similarity metric is used to generate a hierarchy of components automatically through adaptive cluster. As a result, a binary tree is used to represent the hierarchy, in which low level segments are patch-aware regions while high level segments are part-aware components. Therefore, the combination and decomposition relations are clear between segments. Our method is designed to handle arbitrary models, such as CAD model, scanned object, organic shape, large-scale mesh and noisy model. A large number of experiments demonstrate the efficiency of our algorithm.

1. Introduction

The shape analysis and understanding is the key and difficult issue in computer graphics and computer vision fields. The key techniques of the shape understanding are the feature abstract and definition of feature semantics of shape, in which the important step is shape segmentation. The shape segmentation is also widely applied in CAD, animation, scene editing and shape match and compression and so on, especially the part reuse of shape in CAD can increase the efficiency of product design and use ratio of product knowledge. Segmentation has been a well researched in recent years. Survey [1] distinguished between patch-based and part-based methods which are considered to be inherently different. Therefore, few works apply these two types simultaneously. Goes et al. [2] introduced an abstraction that conveys both the perceptual and the geometric structure. But it only gets abstraction of shapes.

With the development of distributed virtual reality system, streaming 3D model technology is imminent. The representation of the models in the form of levels of detail (LODs) is a key technology. But the study on segmentation to generate LODs of models is still blank. In this paper we present a hierarchical splat clustering method based on a novel similarity metric, which combines patch-aware similarity and part-aware similarity. It generates both patch level and part level segments at different levels in the final hierarchy. Therefore, we are able to get LODs of the model, which is shown in Fig. 1. Moreover, large-scale meshes are easier to obtained in recent years. How to segment those models fast is of paramount important for many works. Experiments demonstrate the efficiency of our method, especially for large-scale noise model.

Our method consists of two main steps. First, we decompose the object into splats with an improved VSA (variational shape approximation) method based on an optimized $L^{2,1}$ metric. Second, we merge neighboring clusters with similar metric hierarchically to produce a hierarchy of regions. An overview of the full algorithm is shown in Fig. 2. Main contributions are shown below. Firstly, we introduce an optimized $L^{2,1}$ metric for VSA method. Secondly, we propose a new local patch-aware similarity metric based on splats, which does well in detecting the local feature on the surface of the model. Besides, we define an improved SDF (shape diameter function) of splats, which adopts anisotropic smoothing, as part-aware similarity metric. Finally, the local and global similarities are adaptively combined into a uniform metric, which is used in hierarchical clustering framework. Therefore, we can obtain different levels of segments and their decomposition relations simultaneously.

2. Related works

Mesh segmentation becomes a key ingredient in many computer graphics applications and is surveyed in detail by [1]. Previous algorithms can be categorized into two basic classes based on the types of models they aim to segment, patch-based segmentation and part-based segmentation. To motivate our approach, we present a brief overview of them and the hierarchical face clustering method which is similar as ours.

* Corresponding author
E-mail address: wgp@pku.edu.cn (G. Wang).

http://dx.doi.org/10.1016/j.cag.2015.05.012
0097-8493/

2.1. Patch-based segmentation

There are several algorithms for patch-based 3D mesh segmentation. The traditional algorithms cluster geometric elements with similar properties [1], such as region growing [3–5], hierarchical clustering (see Section 2.3) and spectral analysis [6,7]. These properties include simple surface measures like area, size or length, various differential properties such as curvature, normal direction and some special features like curvature tensor [5]. With a different view of traditional methods, some segmentation methods treat segmentation as an energy minimization problem. Because of its optimization nature, these methods are often referred as variational approaches. The planarity and development ability of surface regions are usually used as error metrics to define energy functions [8–12]. These patch-based algorithms in common are to segment the 3D mesh into regions which represent distinct surfaces. These surfaces can be approximated by various primitives like planes, cylinders, spheres and polynomials.

2.2. Part-based segmentation

Psychological researches [13,14] indicate that human perception tends to cut a shape along concave regions in the direction of minimum principal curvature. Many existing mesh segmentation methods leverage the surface concavity information as a key measure for the underlying algorithms, such as, K-mean clustering [15], graph cut-based fuzzy clustering [16], random walk algorithm [17] and spectral clustering methods [18,19]. There are also some algorithms incorporating some forms of the minima rule to segment the surface of models, such as the approaches based on randomized cuts [20], variational decomposition [21] and concavity-aware fields [22].

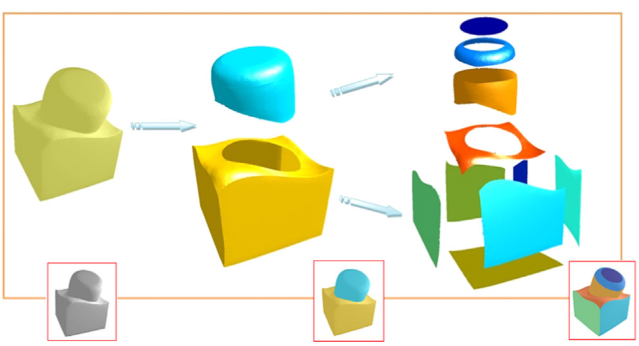

Fig. 1. A simple example of decomposition relationship.

A few works also make direct use of volumetric information for segmentation, such as the shape diameter function [23], the part-aware metric [24], the approximate Convexity Analysis [25] and the continuous visibility feature [26]. The maturity of research on segmentation has culminated in a benchmark which enabled the comparison of different segmentation algorithms [27]. A hierarchical work [28] segments articulated bodies into a coarse-to-fine hierarchy of segments. The goal is as similar as ours, but it is only handle man-made shapes.

Besides those single shape segmentations, there are rising many data-drive co-segmentation [29–31], joint segmentation [32] and supervised segmentations [33,34]. These methods require a training set of labeled shapes, user input, or a set of shapes from the same category. Our goal is to segment individual shapes, which is important when the category of the shapes is unknown in advance, which cannot be clearly categorized or no training set is available.

2.3. Hierarchy face clustering

Hierarchical clustering method merges the pair of regions from the bottom to top hierarchically [35–37,12]. At the beginning, each face of the mesh is assigned as a single region. Then, a pair of adjacent regions with least merging error is merged to form a new region iteratively. This merging is repeated until it meets some stopping criteria. Hierarchical clustering approach generates a binary tree of clusters, which indicates the decomposition relations between components. Our paper improves the hierarchical clustering framework by using splats as the initial clusters, which avoids the influence of noise and greatly improve efficiency on big-data models.

Compared with the previous methods, our method is designed to handle the arbitrary shapes-sharp or smooth, accurate or noisy, sparse or dense. For each specific type of the model and the corresponding method, result analysis of experiment demonstrates the effectiveness of our method.

3. Hierarchical splat clustering framework

We introduce a hierarchical splat clustering framework inspired by traditional HFC (hierarchical face clustering) method [12]. The basic idea is that neighboring clusters merge into representative region by using some passive priority strategies, which defined by an novel similarity metric in Section 4. Differently from traditional HFC method, we use splats as the initial clusters. This can significantly

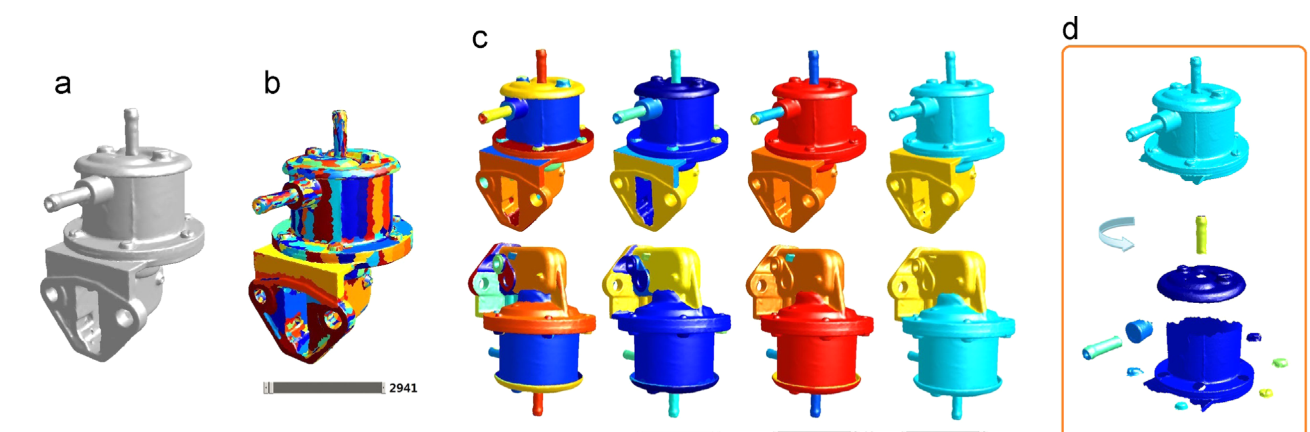

Fig. 2. Overview of our method: (a) a mesh model is given. (b) We decompose this model into 2941 splats in the first step. (c) Patch-aware and part-aware results are obtained at different levels of the hierarchy. (d) An example of one decomposition relation.

reduce the levels of the hierarchical tree and improve the efficiency of the algorithm. The remainder of this section focuses on the definition of splats and how to get them in the first step. More details of HFC method can be referenced in paper [12].

3.1. Splat definition

By the observation on the output planes of the VSA by $L^{2,1}$ metric [8], we find that these planes always show some narrow shape characteristics depending on the surface local curvature (see Fig. 3), we denote them as "Splat". These characteristics guide us to define the local similarity metric well. Besides, splats representation is actually a simplified form of the model. Using splats as clustering atoms can avoid influence of local noise, and have an advantage to handle dense, big-data models.

3.2. Improved VSA by $L^{2,1}$ metrics

This paper introduces an improved $L^{2,1}$ error metric for planes: $\mathbf{E}_n(t_i) = |\boldsymbol{n} - \boldsymbol{n}_i'| \cdot |t_i|$. Instead of triangle normal $\boldsymbol{n}_i$, it measures difference between the optimized normal $\boldsymbol{n}_i'$ on mesh and the plane normal $\boldsymbol{n}$. We use a bilateral filter to smooth the normal field defined over the input mesh based on the method in [38] in the processing. This strategy avoids the adverse effects of noise introduced, which can be omitted for no noise model.

The natural of VSA method is formulating the segmentation as an optimization problem under certain segment number and solving it by a variational approach. But determining this number as prior is hard for arbitrary shapes with different scales. Our optimization problem is converted to obtain the minimum energy and get the number of clusters automatically with constraints of maximal error for each patch. The specific change is that we use an error threshold ϑ (we use $(\pi/15)A_{|t|}$, where $A_{|t|}$ is average area of triangles). First, we use region growing process to get initial planes: seeds are picked randomly to grow regions under error threshold ϑ. The initial step results in a good guess of the unknown planes. In the second step, we assign all the triangles to their nearest planes to drive down the total error by an iterative distortion-minimizing flooding algorithm as in [8]. To quickly obtain splats, all experiments only iterate five times. Fig. 4 illustrates our improved VSA by $L^{2,1}$ metrics, which generate better splats on the noisy oil-pump model compared to traditional method.

4. Similarity metric

We propose a new combination similarity metric based on splats, which distinguish our work from the previous work permanently. It combines two measures: patch-aware local similarity metric and part-aware global similarity metric. Above all, it uses an adaptive integration to intelligent judge which similarity should be dominated in relevant clustering level.

4.1. Patch-aware similarity metric

Splats show local surface features, as in Fig. 3, the area of splat is proportional to the mean curvature; the long and narrow shape has connection with the radio between maximum curvature and minimum curvature; the direction of splat is consistent with the principal direction corresponding to minimum curvature. To describe the main points of the splat patch-aware similarity metric, it is convenient to fix some notations, as shown in Fig. 5. Let splat $\mathbb{S}_i$ is bounded by oriented bounding box $OBB(S_i)$. The means of relevant attributes in this figure are as following: $\vec{n_i}$ is the normal of $\mathbb{S}_i$; O_i is the center of

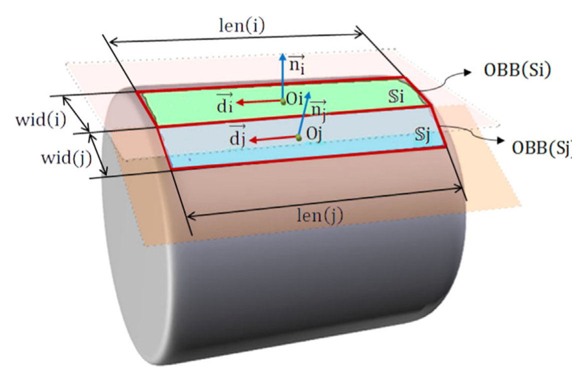

Fig. 5. Two splats $\mathbb{S}_i$, $\mathbb{S}_j$ and their relevant parameters.

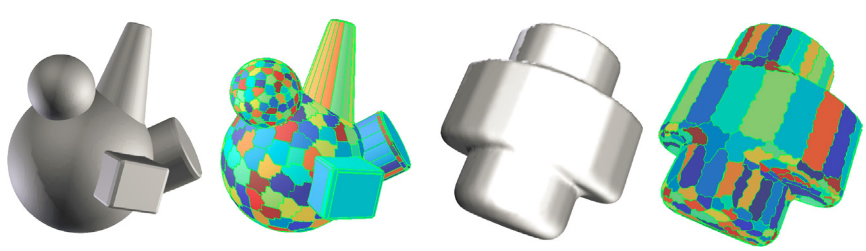

Fig. 3. Results of VSA by $L^{2,1}$ error metric.

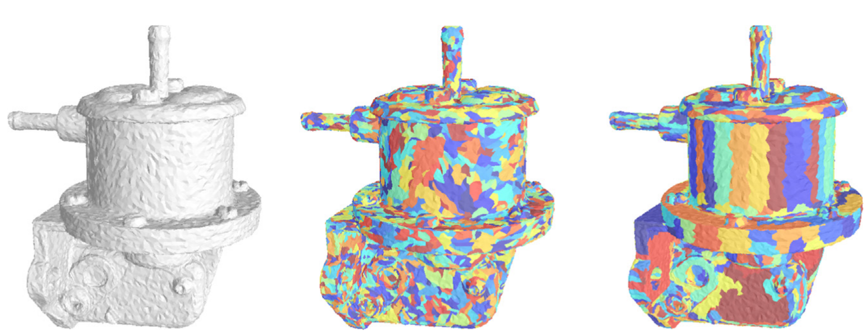

Fig. 4. Splats obtains with VSA by $L^{2,1}$ metric or our improved $L^{2,1}$ metric on noisy model.

$OBB(\mathbb{S}_i)$; $len(i)$, $wid(i)$ are the length and the width of $OBB(\mathbb{S}_i)$, respectively, and the $\overrightarrow{d_i}$ means the long and narrow direction of $\mathbb{S}_i$, which is the direction of the longest side of $OBB(\mathbb{S}_i)$.

In order to faithfully measure this similarity, we use the following elements as basic factors:

- Area similarity $S_a(i,j) = \mathrm{Min}(A_i, A_j)/\mathrm{Max}(A_i, A_j)$, A_i, A_j are areas of $\mathbb{S}_i$ and $\mathbb{S}_j$.
- Shape similarity $S_s(i,j) = \mathrm{Min}(P_i, P_j)/\mathrm{Max}(P_i, P_j)$, P_i, P_j mean long and narrow extend of $\mathbb{S}_i$ and $\mathbb{S}_j$, respectively, which are calculated by $P_i = len(i)/wid(i)$.
- Long and narrow direction similarity $S_d(i,j) = |\overrightarrow{d_i} \cdot \overrightarrow{d_j}|$, where $|\overrightarrow{d_i}| = 1$, $|\overrightarrow{d_j}| = 1$.
- Normal similarity $S_n(i,j) = \overrightarrow{n_i} \cdot \overrightarrow{n_j}$, where $|\overrightarrow{n_i}| = 1$, $|\overrightarrow{n_j}| = 1$.

Considering these factors, the shape similarity between two adjacent splats is defined as

$$S_{shape}(i,j) = \frac{w_1(S_a(i,j) + S_s(i,j) + S_d(i,j)) + w_2 S_n(i,j)}{3 \cdot w_1 + w_2} \tag{1}$$

In our implementation, $w_1 = \sin(\alpha)$ and $w_2 = 1$, where α is the angle between $\mathbb{S}_i$ and $\mathbb{S}_j$. Typically, the factors $S_a(i,j)$, $S_s(i,j)$, $S_d(i,j)$ of two splats on a very flat region do not have any guiding significance. So, the weight parameter w_1 increases as the angle α.

Fig. 6 demonstrates local similarity values associated with various splat pairs. The range of this metric value is 0–1. Fig. 7 shows the visualization of local similarity computed between adjacent splats. It can be noticed that the local similarity is intuitive, and can easily be interpreted visually.

After merging in our framework, one cluster becomes bigger and contains multi-splats. Shape similarity metric between two clusters needs to recalculate. Properties considered in Eq. 1 are local and do not have transitivity. Only splat pairs on the common edge participate in the definition of clusters shape similarity:

$$S_{shape}(I,J) = \frac{1}{n} \sum_{(i,j) \in E} S_{local}(i,j) \tag{2}$$

where E is a set of splat pairs which lay on the common edge of cluster I and cluster J, $n = |E|$.

Besides cluster shape similarity, we introduce a cluster shape optimization factor $O_s(I,J)$ into the finally patch-aware similarity metric to avoid some unsatisfactory merging like Fig. 8:

$$O_s(I,J) = e^{-2\ \mathrm{Min}(A(I),A(J))/C_{Len(I,J)}} \tag{3}$$

where $A(I)$ is the area of cluster I, $C_{Len(I,J)}$ is the length of common boundary. Then local patch-aware similarity metric is defined as

$$S_{local}(I,J) = a \cdot S_{shape}(I,J) + b \cdot O_s(I,J) \tag{4}$$

where the parameters a and b are assigned to be 0.7 and 0.3, respectively in our experiment.

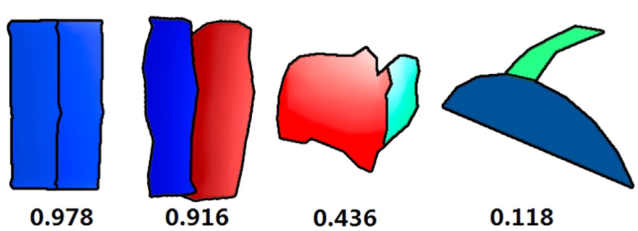

Fig. 6. The local similarity rank associated with various splat pairs.

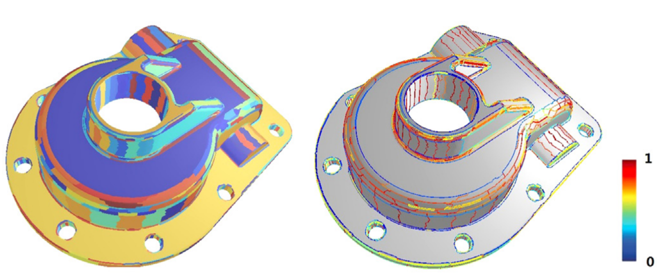

Fig. 7. Visualization of local similarity computed between adjacent splats (red: high similarity, blue: low similarity). (For interpretation of the references to color in this figure caption, the reader is referred to the web version of this paper.)

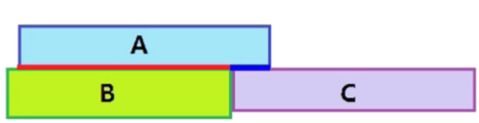

Fig. 8. The red line is the common boundary between A and B, and the blue one is between A and C. Though splat C has higher similarity value with A, the best choice is B because the red line is longer than the blue one. (For interpretation of the references to color in this figure caption, the reader is referred to the web version of this paper.)

4.2. Part-aware similarity metric

As the concept of a "part" is not well-defined, the definition of a part-aware similarity metric is challenging. SDF [23] of triangles provide a connection between the surface mesh and the volume of the subtended 3D bounded object, which are more closely linked to skeletal shape representations.

We calculate splat SDF values based on triangle SDF definition. For each splat, a constant number S_n of triangles ($S_n = 10$ in our implementation) is uniformly subsampled and the SDF value for each triangle is calculated by [23]. Then the initial SDF value of this splat is computed by averaging those triangle SDF values. We set the default values to an opening angle of 90° and send 30 rays per face for triangle SDF calculation. To overcome the effects of local noise, we perform smoothing based on a small number of bilateral filtering iterations of the SDF values on neighborhoods around each splat:

$$f_v^0(x) = f_v(x)$$

$$f_v^{i+1}(x) = \frac{f_v^i(x) + \sum_y c_{x,y} A_y f_v^i(y)}{1 + \sum_y c_{x,y} A_y} \tag{5}$$

where $f_v^i(x)$ is the SDF value of splat x in iteration $i+1$. It is calculated as the average sum of all y neighbors of x, weighted using the area of splat y, and controlled by anisotropy function $c_{x,y}$. If the angle between their normals forms a convexity $c_{x,y} = 1$, otherwise $c_{x,y} = 0$, here we judge whether the angle is convexity or concavity following [25]. Fig. 9 shows the comparison between our method and traditional SDF method [23]. Our method can distinguish the details better than traditional SDF method. For example, the bolts of model oil-pump (the second and the third columns in Fig. 9) are distinguished by our work. And our method gets similar satisfactory result for smooth organic model. Our splat SDF values also have pose-invariant (Fig. 10). In addition, the calculation of SDF includes a mass of intersection ray casting, which leads to high time complexity. Our splat SDF method only computes SDF values of subsampled triangles, so we can achieve the real-time calculation even for big noisy models, like Fig. 18.

We define cluster SDF value $f_{SDF}(I)$ as an weighted average of splats SDF values. The splat's area is used as weight. The part-aware similarity metric between two clusters is defined as the rate of two SDF values:

$$S_{global}(I,J) = \frac{\mathrm{Min}(f_{SDF}(I), f_{SDF}(J))}{\mathrm{Max}(f_{SDF}(I), f_{SDF}(J))}. \tag{6}$$

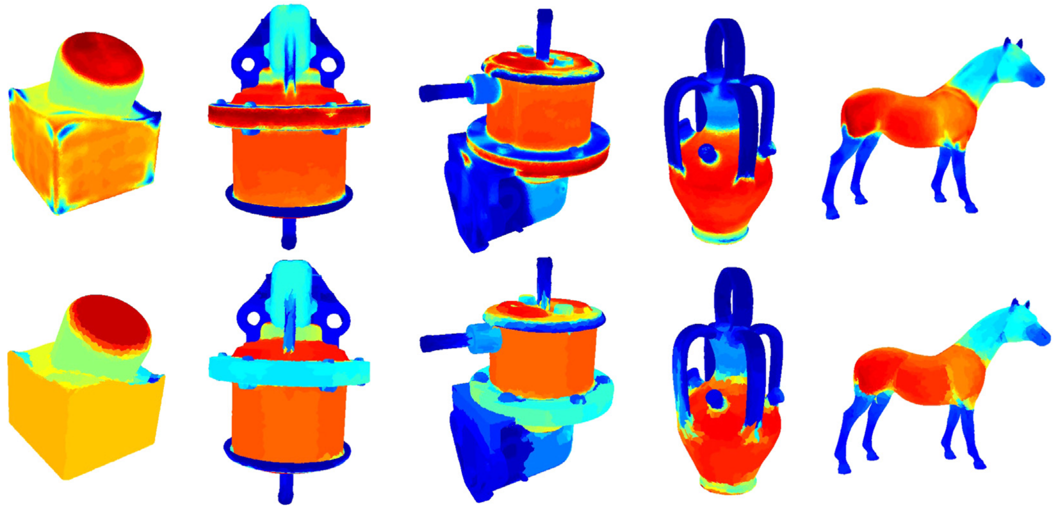

Fig. 9. Comparison between traditional SDF method [23]and our method. First line shows triangle SDF values and the second line is ours. (The traditional SDF values are based on triangles of mesh, which apparent is clear than our splat SDF method.)

Fig. 10. The SDF values of splats are pose-invariant.

4.3. Combination metrics

With the patch-aware metric $S_{local}(I,J)$ and part-aware metric $S_{global}(I,J)$, we use an adaptive pattern to build combined similarity metric $S(I,J)$ between two cluster I,J as

$$S_{local}(I,J) = \alpha \cdot S_{local}(I,J) + \beta \cdot S_{global} I,J \quad (7)$$

where the parameters, α and β control the relative importance of the two metrics. In the first few merging of our hierarchical framework, we are primarily interested in patch level structure, therefore we should emphasize the patch-aware metric. For high level merging, the importance of the part-aware metric ought to be increased. For this reason, we set $\alpha = \sin^2(\theta)$ and $\beta = \cos^2(\theta)$, where θ is an average dihedral angle of the common edge shared by two clusters. By this setting, the global part-aware similarity metric plays a more important role with increasing angle θ.

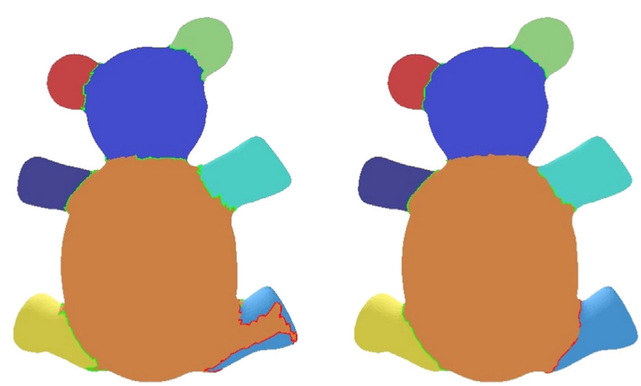

Fig. 11. The red line is smoothed by graph cut smoothing. (For interpretation of the references to color in this figure caption, the reader is referred to the web version of this paper.)

5. Implementation

Implementation details of the presented algorithm are present in this section. Since the calculation of SDF needs consistently normals, we unify normals in the preprocessing. After all splats are received, we compute $OBB(i)$ using a discrete intersection of 2D rectangle rotation algorithm and store the elements $Len(i)$, $Wid(i)$, $OBB(i)$, $\overrightarrow{n_j}$ and $\overrightarrow{d_j}$ for later use. In addition, we use CGAL package for 3D fast intersection and distance computation for our SDF solution. Our method obtains splats and calculates their similarity metric automatically, but the output of our hierarchical clustering is a binary tree which denotes the different levels of segmentation. Users need to decide which level should be chosen as the final reasonable segment result. We also propose a boundary smoothing in the postprocessing to improve the segmentation quality.

As the final step, we employ a graph cut optimization to refine the boundaries between the segments. We create a label l_i^0 for each triangle t_i, which is signed as $l_i^0 = m$ if i belongs to segment region m, and sign the new label as l_i for t_i after graph cut optimization. The energy function $E(X)$ is defined in a similar way to [39], which contains two terms. We set the data term to strictly preserve each segment, but allow the border to move along a narrow band between segments and improve boundary smoothness. We define the data term formally like Ref. [25]:

$$E_1(x_i) = \begin{cases} 0 & \text{if } l_i^0 = l_i, \\ (dist(B_{mn})/maxdist(B_{mn}))^{1.5} & \text{if } l_i^0 \neq l_i \text{ and } l_i = n. \end{cases} \quad (8)$$

Fig. 12. Segmentations obtained with our algorithm on a collection of organic models. A quantitative evaluation on the full segmentation benchmark with 380 meshes is reported in Fig. 17.

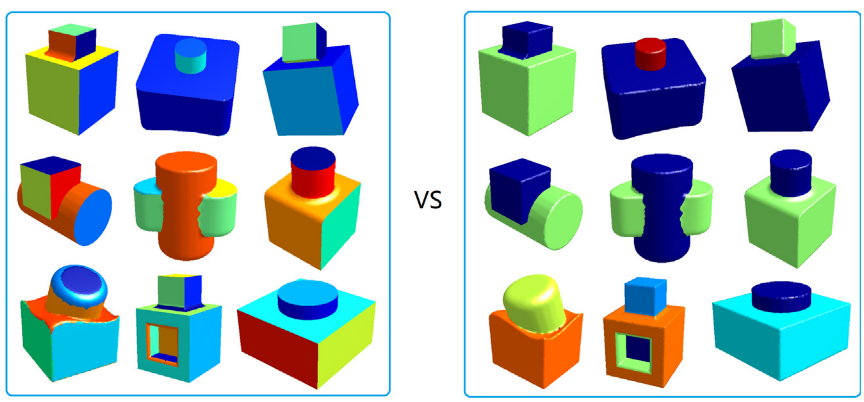

Fig. 13. Patch-aware results and part-aware results on mechanics models.

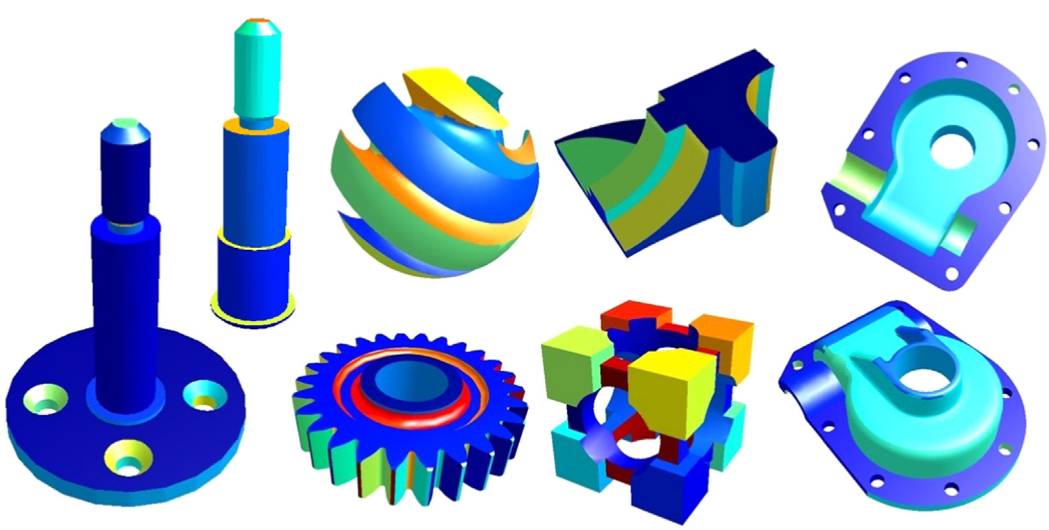

Fig. 14. Segmentations obtained with our algorithm on CAD models.

where $dist(n)$ is the distance of the triangle to the boundary B_{mn} between the segment with labels m and n. The power 1.5 increases the cost faster for triangle away from the boundary. The smoothness term is defined as

$$E_2(x_i, x_j) = E_{normal}(x_i, x_j) + E_{EdgeL}(x_i, x_j) \tag{9}$$

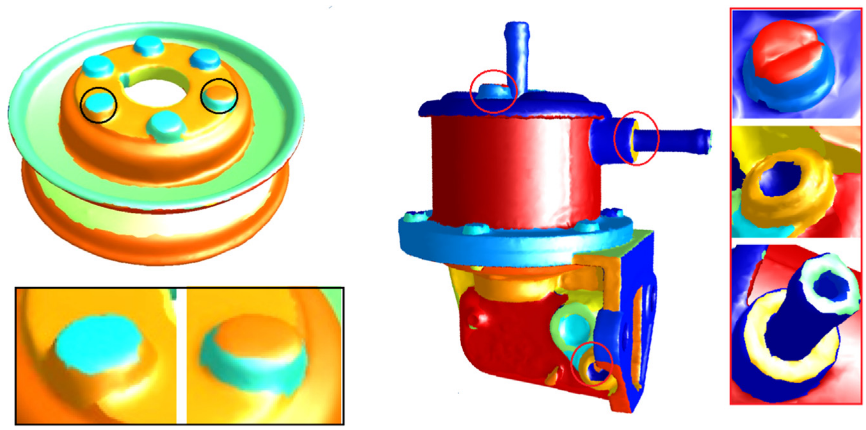

Fig. 15. Segmentations obtained with our algorithm on scanned objects.

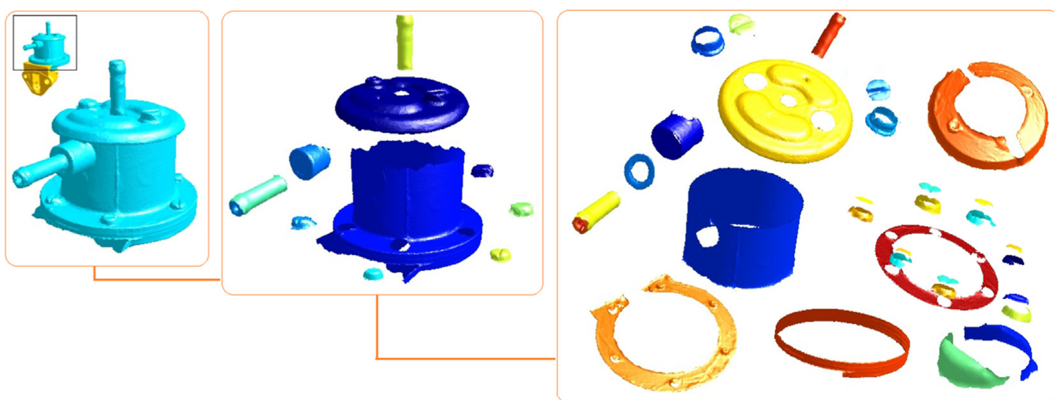

Fig. 16. Illustration of a decomposition relation of model oil-pump.

where $E_{normal}(x_i, x_j) = |\cos(\alpha)|$ and $E_{EdgeL}(x_i, x_j) = len(i,j)/(len(i,j) + aveLen)$ if triangle t_i and t_j are divided into different segment after process; otherwise, $E_{normal}(x_i, x_j) = 1 - |\cos(\alpha)|$ and $E_{EdgeL}(x_i, x_j) = 0$. Here, $len(i,j)$ is the length of the common edge shared by triangle t_i and t_j, $aveLen$ is the average edge length of the mesh. Clearly, $E_2(x_i, x_j)$ ensures the smoothness of the boundary and allows the border to move along a narrow band (see Fig. 11).

6. Experimental results

The hierarchical segmentation presented in this paper is suitable for wide applicable scope. The patch-based or part-based similarity can be used separately. Fig. 13 illustrates examples by using the patch-metric or part-metric alone. Besides, the combination metric ensures to get a hierarchical relationship (as Figs. 1 and 16). Result analysis for each specific type of model is presented in this section.

CAD models: Fig. 14 presents visual examples of segmentation results of CAD models, where many shapes have sharp feature edge. So, this type model is easy to segment. However, great majority of CAD models are obtained by reconstruction of scanned point clouds, which typically are corrupted with noise and outlier. It is difficult to distinguish noises and features. Fig. 15 shows segmentation results of scanned objects. Compared to traditional segmentation methods, the advantage of this paper is that the decomposition relations are implied in the hierarchy. Fig. 16 shows a more detailed example of decomposition relations.

Organic models: The biggest advantage of this method is that it can get part-based segmentation meanwhile. Fig. 12 presents segmentations obtained with our algorithm on a collection of organic meshes. Those models are selected from benchmark of [27]. Fig. 17 shows a comparison of the algorithms according to the various measures of the benchmark. From the results in Fig. 17, we can see that our algorithm is comparable to the state-of-the-art, achieving results similar to shape diameter function and randomized cuts, while also having the advantage of being applicable to get patch-based segment.

Our algorithm only considers the similarity of neighboring regions. So, it pays more attention on tiny feature. Fig. 19 shows flange segment result, in which gear is preserved in the merging process. More examples in Fig. 12 like armadillo's finger, human's ear and nose can confirm this behavior.

6.1. Efficiency

All examples are tested on a PC with Intel core i5 with 2.53 GHz CPU and 2.00 GB RAM. Table 1 presents runtime of our method. To speed up the whole clustering process, the splat SDF values are pre-computed and stored for later use. HC time in this table means time required to compute across the whole hierarchy for models. It is noticed that even though the triangle numbers increase greatly, the total time grows slowly because we use splats as the basic elements of clustering. Fig. 18 shows an example of a noisy big city with more than 100w triangles. It can also get result under 10 s.

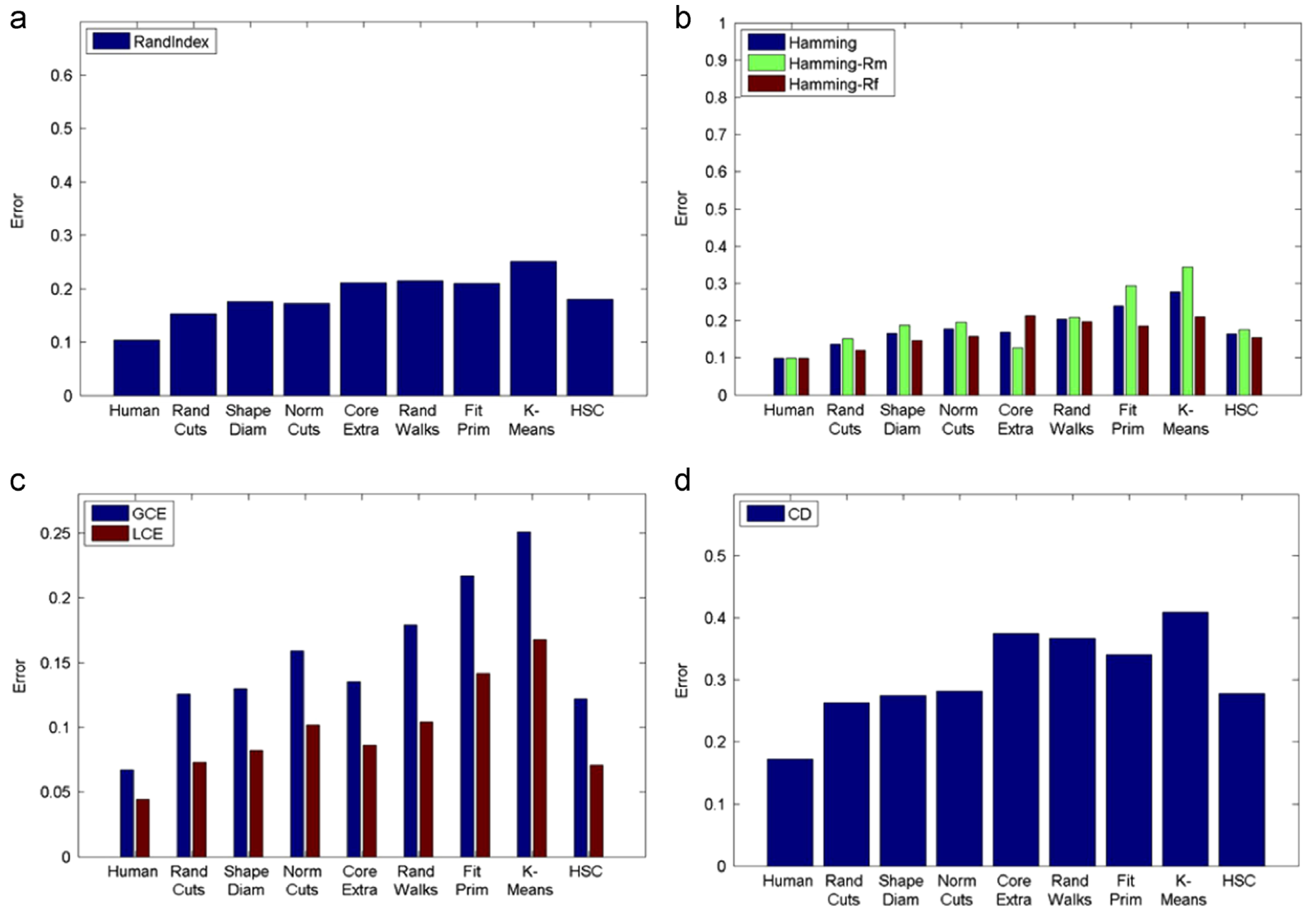

Fig. 17. Quantitative evaluation of our segmentation algorithm on the benchmark of [27]. Our algorithm is named HSC (hierarchical splat clustering).

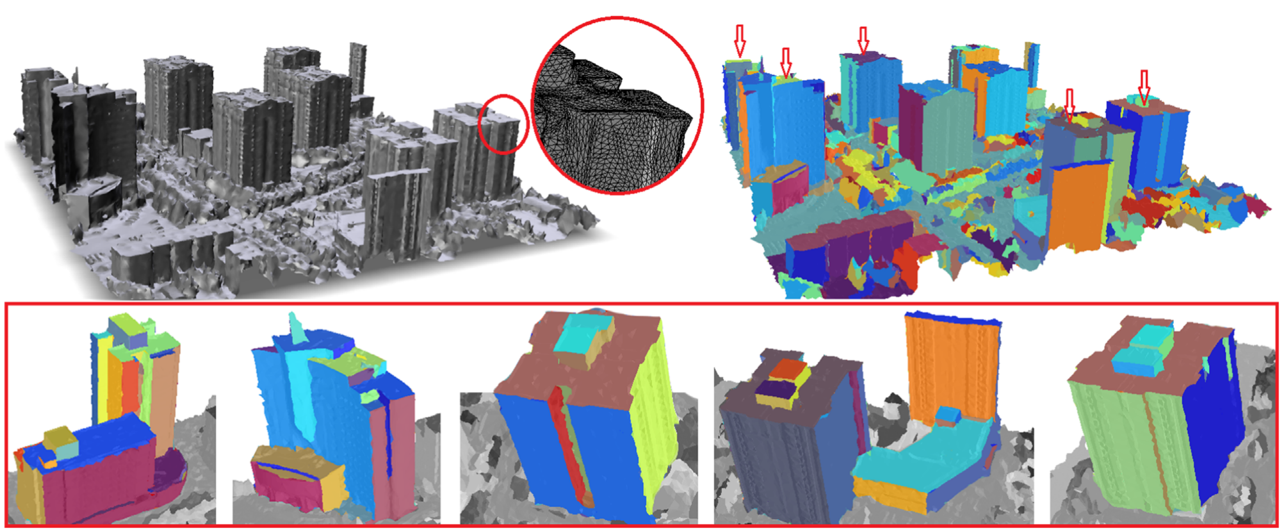

Fig. 18. Segmentations obtained with our method on a big noisy city model.

6.2. Limitation

One limitation of our approach is that it do not suit for all semantic parts adhere to our part characterization. As shown in the example in Fig. 20, some structure is unable distinguished by splat similarity. In addition, we pay more attention to details. Sometimes, we have to retain some fine features in order to obtain a reasonable division. Interactive merging tool is used to remove these subtle features in rare cases.

7. Conclusions and future work

We present a hierarchical clustering segmentation algorithm that uses a novel similarity metric to combine patch-aware similarity and part-aware similarity. The main frame divides the object into a hierarchy. Thus, we could generate patch regions and part components in one hierarchical clustering process. The component decomposition relations are implicit in this hierarchy. We also use a graph cut optimization to optimize the segmentation boundary. We

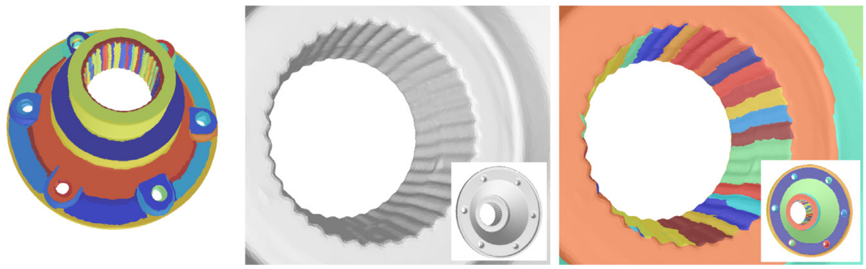

Fig. 19. Details of our segmentations of model flange. It is noticed that our method catch more tiny features.

Table 1
Runtime of our method.

Model	Face numbers	Splat numbers	VSA (s)	SDF (s)	HC (s)
Teddy	27,648	1134	0.889	4.389	0.298
Casting	10,204	498	0.615	3.012	0.205
Flange	66,691	4119	1.030	5.98	0.506
OilPump	79,780	941	0.890	3.88	0.349
Armadillo	345,944	4112	2.013	6.415	0.589
Bigcity	1,051,986	4634	2.623	6.75	0.62

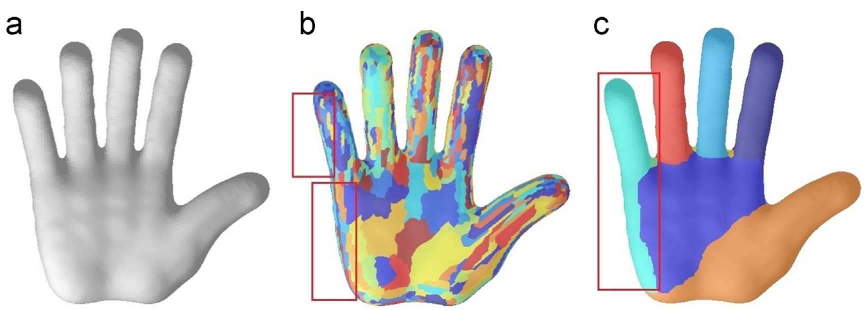

Fig. 20. Limitations of our approach: (a) hand model. (b) splats of hand models. (c) unsatisfactory result.

are able to process arbitrary shapes, regardless whether the shape is sharp or smooth, accurate or noisy, sparse or dense. Because the presented method does not need complicated topological arithmetic, we could realize the other versions which apply for point clouds in the future.

Acknowledgements

The model in this paper is courtesy by the AIM@SHAPE Shape Repository and the Princeton Benchmark. The CGAL package for 3D fast intersection and distance computation is used in our solution of SDF values. This research was supported by Grant nos. 61232014, 61421062, 61170205, 61173080 and 61472010 from National Natural Science Foundation of China.

Appendix A. Supplementary data

Supplementary data associated with this paper can be found in the online version at http://dx.doi.org/10.1016/j.cag.2015.05.012.

References

[1] Shamir A. A survey on mesh segmentation techniques. Comput Graph Forum 2008;27(6):1539–56.
[2] Goes FD, Goldenstein S, Desbrun M, Velhoc L. Exoskeleton: curve network abstraction for 3D shapes. Comput Graph 2011;35(1):112–21.
[3] Choi HI, Choi SW, Moon HP. Mathematical theory of medial axis transform. Pac J Math 1997;181(1):57–88.
[4] Kalvin A, Raylor R. Mesh simplification with bounded error. IEEE Comput Graph Appl 1996;16:3.
[5] Lavoué G, Dupont F, Baskurt A. A new CAD mesh segmentation method, based on curvature tensor analysis. Comput-Aided Des 2005;37(10):975–87.
[6] Liu R, Zhang H. Segmentation of 3D meshes through spectral clustering. In: Proceedings of the 12th Pacific conference on computer graphics and applications, PG 2004. Seoul, Korea: IEEE; 2004. p. 298–305.
[7] Liu R, Zhang H. Mesh segmentation via spectral embedding and contour analysis. Comput Graph Forum 2007;26(3):385–94.
[8] Cohen-Steiner D, Alliez P, Desbrun M. Variational shape approximation. ACM Trans Graph (TOG) 2004;23(3):905–14.
[9] Julius D, Kraevoy V, Sheffer A. D-Charts: Quasi-developable mesh segmentation. Comput Graph Forum 2005;24(3):581–90.
[10] Yamauchi H, Lee S, Lee Y, Ohtake Y, Belyaev A. Feature sensitive mesh segmentation with mean shift. In: 2005 International Conference on Shape Modeling and Applications. IEEE, Cambridge, MA. 2005. p. 236–43.
[11] Yan DM, Wang W, Liu Y, Yang W. Variational mesh segmentation via quadric surface fitting. Comput-Aided Des 2012;44(11):1072–82.
[12] Attene M, Falcidieno B, Spagnuolo M. Hierarchical mesh segmentation based on fitting primitives. Vis Comput 2006;22(3):181–93.
[13] Hoffman DD, Richards WA. Parts of recognition. Cognition 1984;18(1):65–96.
[14] Hoffman DD, Singh M. Salience of visual parts. Cognition 1997;63(1):29–78.
[15] Shlafman S, Tal A, Katz S. Metamorphosis of polyhedral surfaces using decomposition. Comput Graph Forum 2002;21(3):219–28.
[16] Katz S, Tal A. Hierarchical mesh decomposition using fuzzy clustering and cuts. ACM Transactions on Graphics 2003;22(3):954–61.
[17] Lai YK, Hu SM, Martin RR, et al. Rapid and effective segmentation of 3D models using random walks. Comput Aided Geom Des 2009;26(6):665–79.
[18] Zhang H, Liu R. Mesh segmentation via recursive and visually salient spectral cuts. In: Proceedings of the vision, modeling, and visualization; 2005. p. 429–36.
[19] Liu R, Zhang H. Mesh segmentation via spectral embedding and contour analysis. Comput Graph Forum 2007;26(3):385–94.
[20] Golovinskiy A, Funkhouser T. Randomized cuts for 3D mesh analysis. ACM Trans Graph (TOG) 2008;27(5):145.
[21] Zhang J, Zheng J, Wu C, Cai J. Variational mesh decomposition. ACM Trans Graph (TOG) 2012;31(3):21.

[22] Au OKC, Zheng Y, Chen M, Tai CL. Mesh segmentation with concavity-aware fields. IEEE Trans Vis Comput Graph 2012;18(7):1125–34.
[23] Shapira L, Shamir A, Cohen-Or D. Consistent mesh partitioning and skeletonisation using the shape diameter function. Vis Comput 2008;24(4):249–59.
[24] Liu R, Zhang H, Shamir A, Cohen-Or D. A part-aware surface metric for shape analysis. Comput Graph Forum 2009;28(2):397–406.
[25] Kaick OV, Fish N, Kleiman Y, Asafi S, Cohen-Or D. Shape segmentation by approximate convexity analysis. ACM Trans Graph (TOG) 2014;34(1):4.
[26] Guilin L, Yotam G, Jyh-Ming L. Continuous visibility feature. In: 28th IEEE conference on computer vision and pattern recognition (CVPR). IEEE; June 2015.
[27] Chen X, Golovinskiy A, Funkhouser T. A benchmark for 3D mesh segmentation. ACM Trans Graph (TOG) 2009;28(3):73.
[28] De GF, Goldenstein S, Velho L. A hierarchical segmentation of articulated bodies. Comput Graph Forum 2008;27(5):1349–56.
[29] Sidi O, van Kaick O, Kleiman Y, Zhang H, Cohen-Or D. Unsupervised co-segmentation of a set of shapes via descriptor-space spectral clustering [M]. ACM Transactions on Graphics 2011;30(6):126.
[30] Hu R, Fan L, Liu L. Co-segmentation of 3D shapes via subspace clustering. Comput Graph Forum 2012;31(5):1703–13.
[31] Meng M, Xia J, Luo J, He Y. Unsupervised co-segmentation for 3D shapes using iterative multi-label optimization. Comput-Aided Des 2013;45(2):312–20.
[32] Huang Q, Koltun V, Guibas L. Joint shape segmentation with linear programming. ACM Trans Graph (TOG) 2011;30(6):125.
[33] Simari P, Nowrouzezahrai D, Kalogerakis E, Singh K. Multi-objective shape segmentation and labeling. Comput Graph Forum 2009;28(5):1415–25.
[34] Kalogerakis E, Hertzmann A, Singh K. Learning 3D mesh segmentation and labeling. ACM Trans Graph (TOG) 2010;29(4):102.
[35] Garland M, Willmott A, Heckbert PS. Hierarchical face clustering on polygonal surfaces. In: Proceedings of the 2001 symposium on interactive 3D graphics. NewYork: ACM; 2001. p. 49–58.
[36] Gelfand N, Guibas LJ. Shape segmentation using local slippage analysis. In: Proceedings of the 2004 eurographics/ACM SIGGRAPH symposium on geometry processing. ACM; 2004. p. 214–23.
[37] Marinov M, Kobbelt L. Automatic generation of structure preserving multiresolution models. Comput Graph Forum 2005;24(3):479–86.
[38] Zheng Y, Fu H, Au OKC, et al. Bilateral normal filtering for mesh denoising. IEEE Trans Vis Comput Graph 2011;17(10):1521–30.
[39] Li Y, Sun J, Tang CK, Shum HY. Lazy snapping, In: ACM TOG, SIGGRAPH, vol. 23, issue no. 3; 2004. p. 303–308.