



Improved quadric surfaces recognition from scanned mechanical models

Quan Weize^{1,2}, Guo Jianwei¹, Zhang Xiaopeng¹, Yan Dongming¹

1. National Laboratory of Pattern Recognition, Institute of Automation, Chinese Academy of Sciences, Beijing 100190, China;

2. University of Chinese Academy of Sciences, Beijing 100049, China.

Abstract: This paper presents a novel algorithm for identifying quadric surfaces from scanned mechanical models. We make several important improvements over the existing variational 3D shape segmentation framework, which utilizes Lloyd's iteration. First, instead of using randomized initialization (which likely falls into non-optimal minimum), the RANSAC-based initialization approach is adopted. Given a good initialization, our method converges quickly than previous approaches. Second, in order to enhance the stability and the robustness, we carefully modify the distortion-minimizing flooding algorithm by using seed regions instead of seed triangles. Third, the geometric constraints are introduced into the optimization framework. The segmentation quality is further improved. We validate the efficiency and the robustness of our proposed method on various datasets, and demonstrate that our method outperforms state-of-art approaches.

Key words: shape segmentation; quadric surfaces; geometric constraints; Lloyd's iteration

1 Introduction

Triangulated mesh surface is one of the most important representation of 3D objects in both computer graphics and computer vision. Mesh segmentation can give a semantic description and understanding of 3D shapes, which plays a central role in various applications, such as part-based recognition^[1] and modeling^[2], 3D mesh compression^[3-5], deformation^[6], remeshing^[7-8], and so on.

Mesh segmentation has been gained a lot of focus in the past decades. Most previous segmentation algorithms are based on hierarchical clustering or feature-based techniques. Huber et al.^[1] first proposed the variational shape approximation (VSA) framework to partition 3D mesh into a minimal set of planar patches while minimizing the fitting errors. Wu and Kobbelt^[9] extended VSA framework by allowing for several different primitives (e.g., spheres, and cylinders) to represent the geometric proxy. Other primitives, such as ellipsoidal surfaces^[10], developable patches^[11], general quadrics^[12-13] are also studied.

Inspired by recent works in shape detection of

point clouds^[14-15], we present several improvements of the VSA framework for quadric surfaces recognition from mechanical parts. In our framework, the RANSAC-based method^[16] is applied to initialize the initial partition, instead of traditional strategy that randomly selects a set of triangles for initialization. The advantage of RANSAC-based initialization is that it gives a good initial guess of the final segmentation, and thus can obtain a better segmentation without using region teleportation operator. Furthermore, the VSA framework only considers the local fitting error of each patch, which cannot find the global relationship between patches. Our improved framework can also automatically extract simple geometric constraints to force the segmentation result to follow the salient structure of shapes. The main contributions of this work are as follows:

- (1) Replacing the random strategy with RANSAC-based strategy for initialization.
- (2) Introducing some geometric constraints into VSA framework to improve the quality of segmentation.
- (3) Modifying the distortion-minimizing flooding

algorithm^[7] to enhance the stability and robustness of partitioning.

2 Related work

In this section, we briefly review two types of mesh segmentation algorithms that are closely related to our work, as well as the primitive detection algorithms. More comprehensive discussions of mesh segmentation can be found in survey papers^[17-18].

2.1 Greedy approaches

This category of approaches mainly includes region growing, hierarchical clustering and hierarchical decomposition. In region growing approaches, a set of faces are first selected as seeds of regions, and then those seeds keep growing until all faces of the mesh are assigned to a region. Local surface properties, such as normal and principle curvatures, are usually used to guide the growing process^[19-21]. In hierarchical clustering approaches, each face of the mesh is first regarded as a single region. At each clustering step, two adjacent regions with least merging error are merged to construct a new region^[22-23]. In hierarchical decomposition (or mesh splitting), mesh surfaces are segmented into meaningful components in a top-down manner, which is opposite to hierarchical clustering^[24-26]. The main characteristic of greedy approaches is that a triangle element is assigned to a region and will not be changed in the later process.

2.2 Variational approaches

Another opposite strategy is iterative optimization, also known as variational approach. Huber et al.^[1] proposed a variational geometric partitioning framework for shape approximation. In their approach, the plane is used as basic geometric proxy. A new appropriate error metric $L^{2,1}$ was introduced to measure the total approximation error. Due to the NP-hard nature of this problem, a novel iterative optimization method, which is an efficient extension of Lloyd's algorithm^[27], was developed. Practically, this method involves two steps: geometry partitioning(the original faces are divided into non-overlapping connected regions via a distortion minimization flooding algorithm); proxy fitting(computing an optimal geometric proxy for each above region). This work used only planes as basic primitive, which leaded to produce too many planar regions for segmentation purpose. Several researchers extended this framework via introducing

higher order or special type of geometric primitives, e.g., ellipsoids^[10], simple quadrics such as spheres, cylinders^[9], and general quadric surfaces^[12-13].

2.3 Shape detection from point-cloud

Schnabel et al.^[16] proposed a state-of-the-art RANSAC-based algorithm to detect basic shapes from the input point cloud. This algorithm is robust for outliers and noise because of the nature of RANSAC. However, this approach is scale-sensitive, that is, it is less efficient on the case of input point cloud containing some small shapes. Furthermore, such a local approach can be unreliable, especially in regions of biased noise or incomplete data, and thus lead to global inconsistency. To solve this problem, Li et al.^[14] proposed GlobFit method that starting with the results of RANSAC-based algorithm^[16] and enforce these primitives along with their global mutual relations. GlobFit proceeds from a coarse to fine scale, specifically, performs regularization after complete detection, and then re-detection in the remaining unclaimed points until the remaining points are little or a maximum number of iterations is reached. Differently, Oesau et al.^[15] proposed planar shape detection algorithm that performs detection and regularization in tandem. In our approach, we also utilize RANSAC for initialization purpose.

3 Problem formulation

Given an input mesh $M = \{t_i\}_{i=1}^n$ (where n is the number of triangles), and a desired number k of regions (or clusters), we denote a segmentation of M by $R = \{R_j\}_{j=1}^k$, where $\bigcup_{j=1,\dots,k} R_j = M$ and $R_i \bigcap R_j = \emptyset$ for any $i \neq j$. Then each region R_j is approximated by a geometric proxy P_j . The total approximating error of a segmentation is defined by

$$E(R, P) = \sum_{j=1}^k E(R_j, P_j) = \sum_{j=1}^k \sum_{t_i \in R_j} E(t_i, P_j) \quad (1)$$

where $E(t_i, P_j)$ measures the distance of a triangle to a proxy.

3.1 Triangle-proxy distance

For triangle-proxy distance, we follow the definition given in the work^[13]. A proxy P_j is described by a general quadric surface $f(X) = C^T \cdot F = 0$, where “.” is the matrix product, $X = [x, y, z]^T$ is a 3D point,

$\mathbf{C} = [c_0, c_1, \dots, c_9]^T$ is the coefficient vector, and $\mathbf{F} = [1, x, y, z, x^2, xy, xz, y^2, yz, z^2]^T$ is the same dimensional vector. Similarly, both Euclidean distance and normal deviation are considered for triangle-proxy distance.

$$E(t, P) = E(t, f) = E_{L^2}(t, f) + \beta E_{L^{2,1}}(t, f) \quad (2)$$

where E_{L^2} measures the squared Euclidean distance between triangle t with proxy f , $E_{L^{2,1}}$ calculates the normal deviation from t to f , and β is the balance weight.

$$\begin{aligned} E_{L^2}(t, f) &= \int_t \frac{f(X)^2}{|\nabla f(X)|^2} d\sigma \\ E_{L^{2,1}}(t, f) &= \int_t \left(\frac{\nabla f(X)}{|\nabla f(X)|} - \mathbf{n}_t \right)^2 d\sigma \end{aligned} \quad (3)$$

where \mathbf{n}_t is the unit normal vector of t , and σ is the integral unit.

4 Proposed algorithm

In this section, we describe the implementation details of the presented algorithm. Our segmentation algorithm consists of preprocessing, initialization and constrained optimization. The goal of preprocessing is to unify the scales of different meshes. The purpose of the improved initialization is to give a relative good initial partition, which can lead to faster convergence speed. Note that this initialization cannot ensure the identical number of segments during multiple executions for the same input mesh, due to the random nature of RANSAC. The key step of this algorithm is the iterative optimization that finds a optimal segmentation through energy minimization.

4.1 Preprocessing

In this step, the input mesh M is first uniformly scaled into the unit cube $[0,1]^3$. In order to improve the robustness of proposed algorithm, we also subdivide mesh so that all the edges of M is shorter than the specific length (default=0.03)^[28]. This approach could remove the elongated triangles with long edges, which will influence the accuracy of segmentation.

4.2 Initialization

In classical VSA framework, random strategy is often used to produce the initial configuration. In this work, the RANSAC-based method is adopted for this purpose. First, the dual point set S of M is obtained,

where the barycenter of the triangle face correspond to each point in S . The normal of each point is equal to the normal of its corresponding triangle facet. After RANSAC-based partitioning, we can get an initial segmentation of S , and an initial segmentation of M can also be obtained because the triangles of M and points of S are the one-to-one correspondence relationship. The result of RANSAC is non-connected and leaves some unclassified points, therefore, several further steps is needed to obtain the connected and non-overlapping regions.

(1) Remove non-connected regions. This step needs to traverse all the regions, and for each region R_i , its connectivity is first checked. If it is connected, then this region keeps unchanged; if the region has two connected components, then a new cluster is created; if the region has more than two connected components, and the connectivity ratio of the second largest connected components is larger than 0.25, then the largest and the second largest connected components are reserved; otherwise, only the largest connected component is reserved. After this process, all the clusters are connected.

(2) Process unclassified faces. Through RANSAC, there are some faces have not been assigned to any cluster. In addition, some new unclassified triangles appear after the above operation. A simple and naive strategy that progressively assign these triangles to adjacent clusters is utilized. Consequently, all the triangles in the mesh is partitioned into the connected and non-overlapping regions.

4.3 Iterative optimization

In our approach, the mesh segmentation is formulated as an energy minimization problem, which can be solved via a simple extension of Lloyd's algorithm, i.e., distortion-minimizing flooding algorithm^[7]. This algorithm mainly includes two steps, i.e., partitioning and surface fitting, and they are in tandem. Given the number of segments of the specific mesh, the region merging or region insertion may be needed to adjust the number of partitions because of the random nature of above initialization. Furthermore, the geometric constrained optimization is introduced into surface fitting to improve the quality of the segmentation, especially for the noisy meshes.

(1) Partitioning. Before performing the re-grouping, a seed region is first found for each region of R . In VSA framework, it usually use the triangle with the

smallest fitting error as the seed triangle. However, this approach is not robust for models with noises, and this is why that QSF^[13] is noise sensitive. To improve the robustness, we propose to use seed region instead of seed triangle, which is more stable and robust to noise. In detail, for each cluster R_i , we compute first m ($m=10$) smallest fitting error triangles $T = \{t_i\}_{i=1}^m$, and collect corresponding neighbors $N = \{N_i\}_{i=1}^m$ of these triangles, where N_i is a set of neighboring triangles of t_i . The number of triangles in N_i is equal to $0.1 \times |R_j|$, where $|R_j|$ is the number of triangles in R_j . For each N_j , its area-average fitting error is first calculated, hereafter, N_j with the smallest area-average fitting error is regarded as seed region and is used for following re-grouping.

A global priority queue Q is dynamically maintained during the partition step, whose priority is equal to the distance of the triangle-proxy pair $P(t_i, P_j)$. Each triangle has two properties, i.e., visited (default as ‘false’) and cluster index (default as -1), denoted by $\langle v, idx \rangle$. First, the triangle-proxy distances are computed for the triangles in seed regions of all clusters, and then insert them into Q . For these seed triangles, $v=true$ and idx is set as the index of corresponding proxy. Then, we progressively pop the top element $P(\tilde{t}, \tilde{P})$ of Q with the smallest triangle proxy distance, and check the proxy assignment of triangle \tilde{t} . If $idx \neq -1$, we do nothing; otherwise, idx is set as the index of current proxy \tilde{P} . Afterwards, all the unvisited incident triangles (i.e., $v=false$) of \tilde{t} in the current proxy \tilde{P} are tested, and these new triangle-proxy pairs (up to two) are pushed into Q . The growing process stops when the priority queue is empty, and then a new partition of the input mesh is obtained.

(2) Surface fitting. Once a new cluster is obtained after the above operation, we assign an optimal proxy P_j to each region R_j . The planarity of region R_j is first computed using principle component analysis (PCA)^[29]. The definition of planarity is

$$p(R_j) = \frac{\lambda_1}{\lambda_1 + \lambda_2 + \lambda_3} \quad (4)$$

where $0 \leq \lambda_1 \leq \lambda_2 \leq \lambda_3$ denote the three eigenvalues of the covariance matrix constructed on the vertices and barycenters of all the triangles of R_j . If the planarity is smaller than a threshold (default= 10^{-6}), the region R_j is fitted by a plane. Otherwise, a quadric surface is fitted for this region. Note that we use PCA

to specify whether the region is a plane or not, instead of the absolute fitting error of plane used in the work^[13]. The reason is that PCA is more robust for noisy models.

(3) Region merging. When the number of current regions is larger than κ (the desired number of regions in the model), some pairs of adjacent regions are merged into larger ones. For example, when adjacent regions are both planes, the angle θ between their normals is calculated. If $\theta \leq 5^\circ$, these two regions are considered coplanar and can be merged into a larger plane. Otherwise, the following method is carried out: computing the fitting error E_{ij} for each pairwise adjacent components (R_i, R_j) , and then merging the pair with the smallest increasing error, i.e., $\min_{(R_i, R_j)} |E_{ij} - (E_i + E_j)|$, where E_i and E_j are the fitting errors of region R_i and R_j , respectively. Finally, a new proxy is fitted to this new component.

(4) Region insertion. On the contrary, the region insertion operator is employed when the number of current regions is less than κ or the fitting error of a region is larger than a threshold. A new region will be inserted into the “worst-fitted” region. In detail, we first traverse all the current regions and find the region R_j with the maximal area-average fitting error

$\max_j \left(\frac{E_j}{A_{R_j}} \right)$, where A_{R_j} is the total area of all the triangles in region R_j . Then, the face with the largest fitting error in R_j is regarded as seed face of new region R_{new} , and a planar primitive is used for fitting R_{new} . Differently from the previous work^[1,13], a local re-grouping and surface fitting operator over R_j and R_{new} is applied, instead of the global Lloyd iteration.

(5) Constrained optimization. The mechanical models usually contains many constraints, e.g., parallel/orthogonal planes, coaxal cylinders, and so on. The normals and axes of planes and cylinders might not be strictly parallel or orthogonal after the optimization stage, especially for noisy models. We adapt the constrained optimization algorithm proposed in the work^[14] to enforce the constraints between planes, cylinder, and cones.

5 Experimental results

To evaluate the effectiveness and the performance

of the proposed algorithm, we conduct various experimental results on a set of mechanical models. All the results shown in this paper are tested on a PC with 3.60 GHz i7-4790 Four CPU, 16 GB memory, and a 64-bit Windows 7 operating system.

5.1 Parameters setting

The classical RANSAC-based method^[16] has three free parameters: α is the maximum normal deviation between one point and the shape, ϵ is the Euclidean tolerance error distance of a compatible point, and τ is the minimal shape size. For

convenience, we use empirical values of the first two parameters, i.e., $\alpha=0.9$ and $\epsilon=0.02$. Controlling τ can roughly adjust the number of regions. Fig.1 shows the recognition results of a part with increasing minimal shape sizes. A larger τ generally results in less regions, and larger fitting errors. We choose $\tau=50, 100$ or 200 for all examples shown in this paper. In practice, the user can tune this parameter to obtain a good guess. However, tuning τ cannot precisely control the number of segmented regions. Therefore, another parameter κ (the number of regions in the model) is introduced.

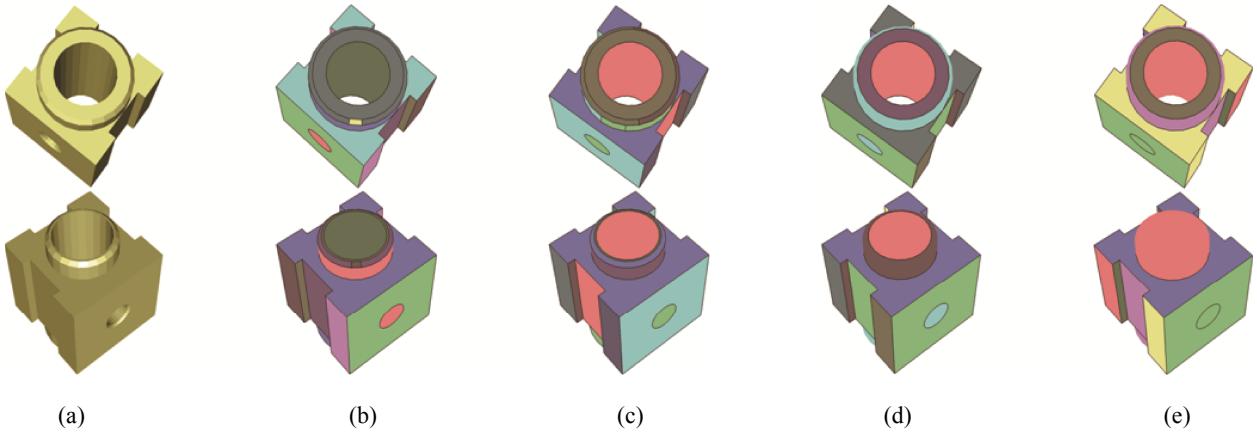


Fig. 1 Segmentation results with increasing minimal shape sizes. Top and bottom rows are two different views of results. (a) Input. (b) $\tau=50$, the number of region is 28. (c) $\tau=100$, the number of region is 28. (d) $\tau=200$, the number of region is 22. (e) $\tau=500$, the number of region is 19.

5.2 Validity

In order to verify the validity of our method, we demonstrate segmentation results of models with various structures and complexities, as shown in Fig.2. It is shown that RANSAC can give good initial guess, especially in the recognition of planar regions. Based on this good initial guess, our method can obtain an improved result quickly and robustly.

In addition, some experiments are done to demonstrate the power of geometric constraints. Random noise is added into the clean models, and then we compare segmentations of those noise models with/without using constraints. The segmentation results are shown in Fig.3. We also statistic three metrics, i.e., fitting error, hamming distance, and consistency error, which are shown in Table 1. The last two metrics have been widely used for quality evaluation of image and shape segmentation^[3]. Adding noise to the input mesh that uses the following strategy: selecting parts of vertices randomly, and then computing the new coordinate for each selected vertex v using Eq.(5).

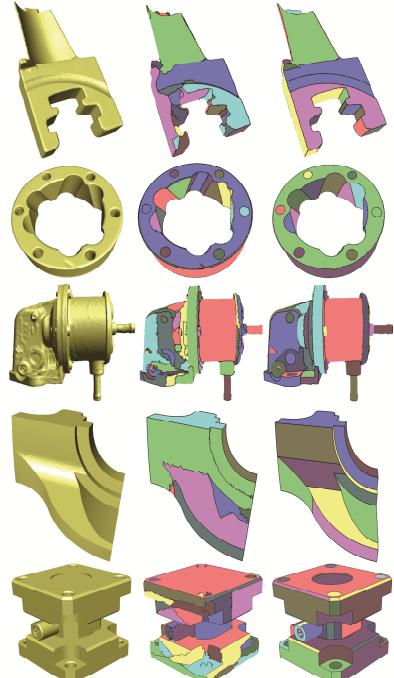


Fig. 2 More segmentation results of our method. Each group includes input model, initial segmentation via RANSAC, and final result. Models are from top to bottom: blade, rolling stage, oil pump, fandisk and part4. Please see Table 3 for the parameters and timings.

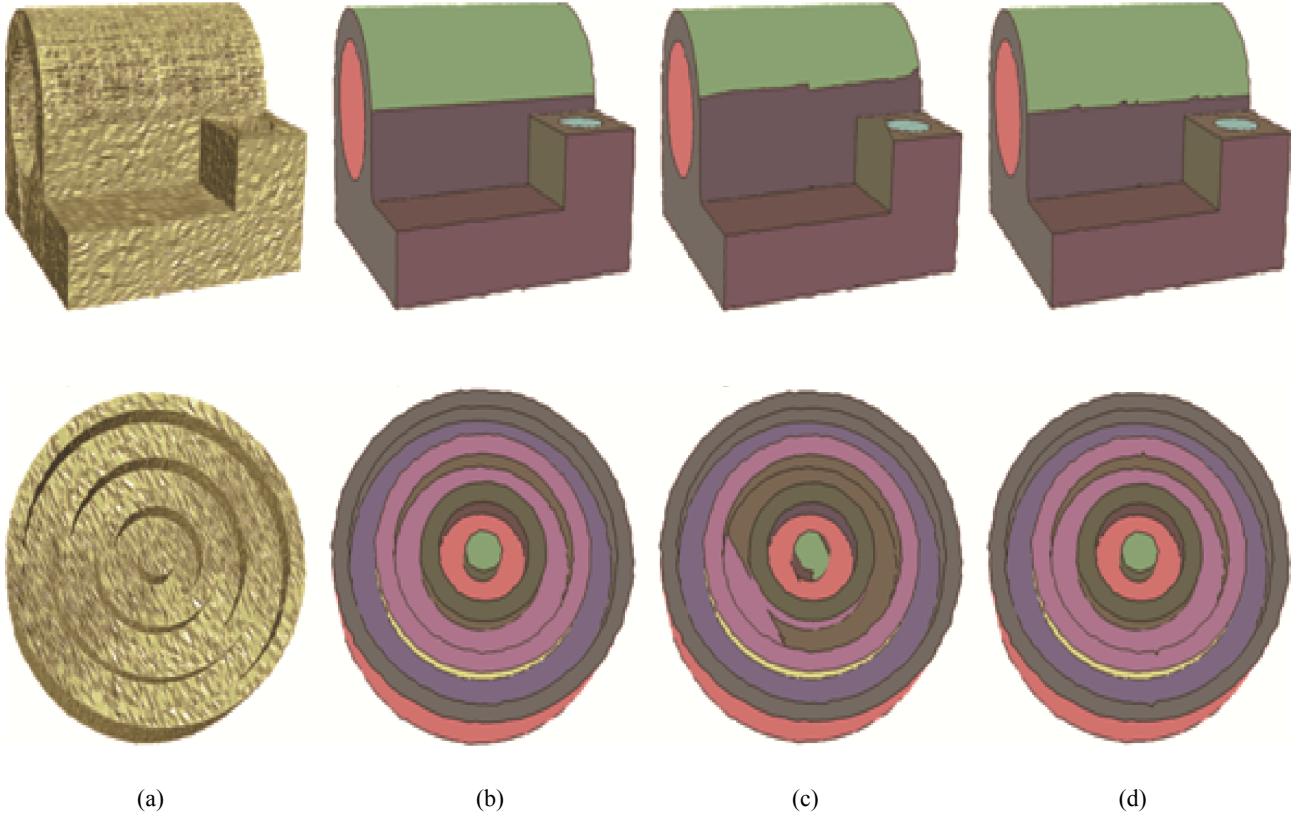


Fig. 3 Comparison of segmentation results with/without using constraints (Noise parameters: $\sigma=0.005$, and the vertex ratio is 50%).
 (a) Input model. (b) Ground truth. (c) Without constraints. (d) With constraints.

Table 1 Statistics of fitting error, HD and CE. For each model, the first row without using constraints, and the second row using constraints.

Model	Error	HD(%)	CE(%)
Joint	1.578	3.88	4.39
	1.572	0.05	0.09
Shell	0.975	2.16	3.97
	0.967	0.007	0.009

$$v_{noise} = v + \sigma^* \omega^* n_v \quad (5)$$

Where σ is the scale factor (default = 0.01) and can be changed, $\omega \in [-1, 1]$ is uniformly distributed random number, and n_v is the unit normal vector of vertex v .

The definitions of hamming distance and consistency error are described in the following. Given two segmentation results $R = \{R_1, R_2, \dots, R_n\}$ and $\hat{R} = \{\hat{R}_1, \hat{R}_2, \dots, \hat{R}_n\}$ with the same number of regions, where \hat{R} is the ground truth. The hamming distance (HD) is defined by

$$HD(R, \hat{R}) = \frac{1}{2|M|} \left(\sum_{i=1}^n \|R_i \setminus \hat{R}_i\| + \sum_{i=1}^n \|\hat{R}_i \setminus R_i\| \right) \quad (6)$$

where $\|\cdot\|$ means the total area of all faces in the corresponding set, and the “\” symbol is the set difference operator. Note that the HD is a symmetric measurement.

Denoting the segment in R that contains face f_i by $S(R, f_i)$, the local refinement error can be defined as $E(R, \hat{R}, f_i) = \frac{\|S(R, f_i) \setminus S(\hat{R}, f_i)\|}{\|S(R, f_i)\|}$. Then, the (Global) consistency error (CE) is defined as

$$CE(R, \hat{R}) = \frac{1}{|M|} \min \left\{ \sum_i E(R, \hat{R}, f_i), \sum_i E(\hat{R}, R, f_i) \right\} \quad (7)$$

where $|M|$ is the number of faces in mesh M .

In our work, the segmentation of the clean models are taken as ground truth to compute CE and HD. The results without using constraints (Fig.3 (c)) will fail in flat region, and be sensitive to noise. Fig.3 (d) can give an almost right result except have few burrs due to very strong interruption. In Table 1, the three indicators (i.e., fitting error, HD and CE) are both decreased, as demonstrated, when using geometric constraints. Especially, the CE and HD are very small (<0.1%) after adding normal constraints, which means our segmentation results over noise models are nearly

consistent with that of corresponding clean models.

Another advantage of introducing geometric constraints into primitive fitting is to improve the segmentation boundary. In Fig.4(a), there are some

zigzag boundaries between clusters. We reprocess the surface fitting subject to normal constraint, and then perform regrouping again. The boundaries are smoothed as shown in Fig.4(b).

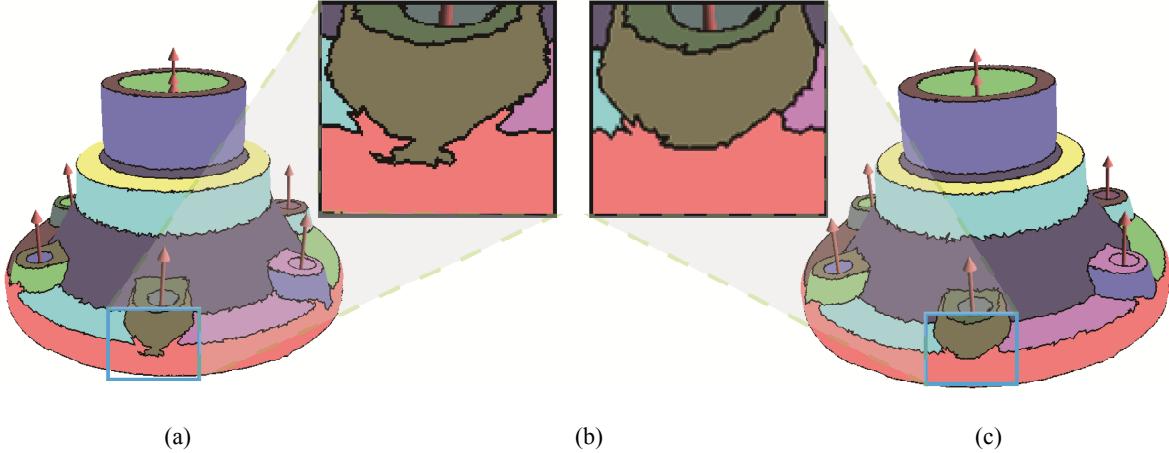


Fig. 4 Improvement of segmentation via geometric constraint. (a) Original result. (b) The closeup of results. (c) Improved result.

5.3 Seed triangle VS seed region

In this work, there is the observation that using seed regions is more stable and robust than using seed triangle for noise models. The comparison of these two strategies is shown in Fig.5. The fitting errors are both decreasing with the iterative process. However, dotted line (using seed triangle) has greater oscillation than solid line (using seed region). In classical VSA framework^[1], the seed triangle is the most similar to its associate proxy, that is, finding the one triangle with the smallest distortion error by traversing all triangles in corresponding region. It can be approximate for only plane primitive. But for general quadric surfaces and noise models, a seed region (i.e., a set of triangles) that is more similar to its proxy is more reasonable for subsequent re-partitioning. And, this example specifies such reasonableness.

5.4 Robustness

Many 3D meshes reconstructed from noise scanned point clouds contain different levels of noise. The robustness of our method is evaluated on several synthetic and real-world noise datasets, by comparing the difference between segmentation result and the ground truth. We use two metrics (i.e., hamming distance and consistency error) to evaluate the quality of segmentation. Practically, the segmentation result of the clean mesh is regarded as the ground truth, and evaluate the segmentation quality of the noise mesh (adding random noise into original clean mesh via Eq.(5).

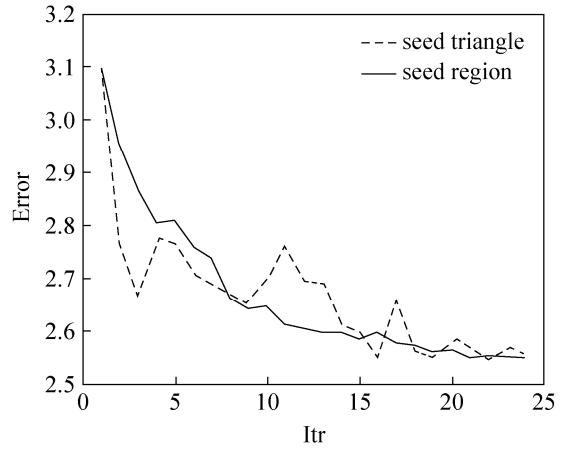


Fig. 5 The chart shows the convergence of different strategies of seed selection for noise “joint” model, in the case of using same initial segmentation.

Given a clean mesh model (“twelve”) and corresponding segmentation result as the ground truth, we randomly select parts of vertices (the vertex ratio is $\{0\%, 10\%, 20\%, \dots, 90\%\}$) and add random noise ($\sigma = 0.01$). Then, those noise models are segmented via our method, and parts of results are shown in Fig.6, where vertex percent of Fig.6(c) and Fig.6(d) are 20% and 80%, respectively. The level of noise only slightly influences the smoothness of segmentation results. Furthermore, the HD and CE of results are counted and shown in Fig.7. With the increase of noise scale, the segmentation result decrease slightly. It is easy to see that the HD and CE is very small ($HD_{max} < 0.04$ and $CE_{max} < 0.08$) even though 90% vertices is contaminated by noise.

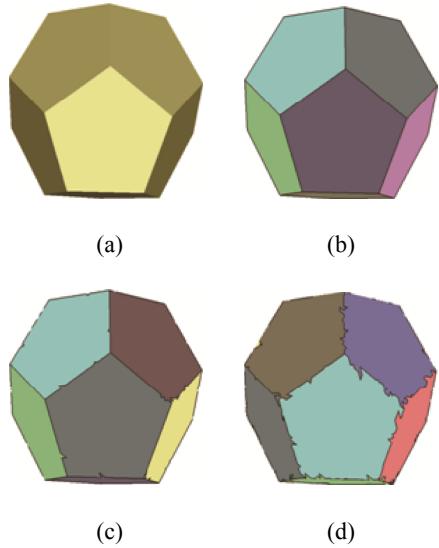


Fig. 6 Segmentation results on the “twelve” model. (a) The input model. (b) Corresponding segmentation result. (c) and (d) Segmentation results on “twelve” model with different amount of noise, respectively.

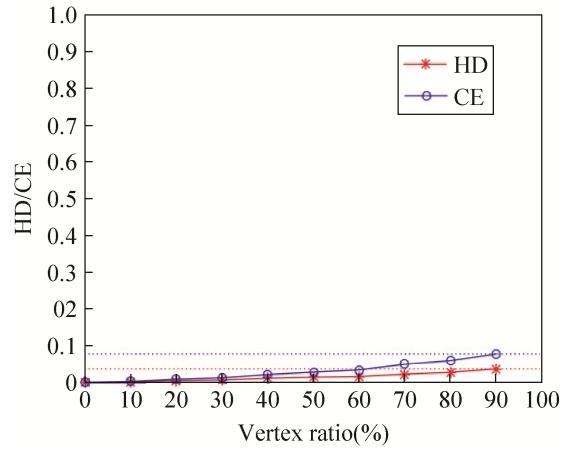


Fig. 7 HD and CE of Noise mesh segmentation. The dot lines correspond to the max value of HD and CE, respectively.

In addition, we apply our method on the real noise models, and the results are shown in Fig.8. Although, the input meshes have some noise (i.e., geometric textures), our method can give a right result.

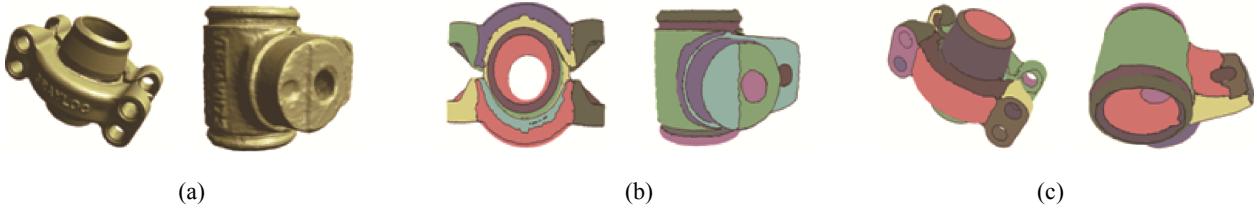


Fig. 8 Noise mesh segmentation. (a) Original model. (b) and (c) are two views of segmentation results.

5.4 Comparison

Fig.9 compares our segmentation results with QSF^[13] using the “bone” model. The running time and fitting errors are shown in Table 2. Due to the better initialization and geometric constraints, the initial fitting error of our method is smaller (see Table 2 (Error 0)) and the segmentation quality is higher (Table 2 (Error 1)). In order to further comparison, the converge rates of our method and QSF^[13] are also shown in Fig.10. Because the initial segmentation roughly capture structure of mesh, our method can converge faster than QSF. Moreover, our method can steadily converge due to the strategy of seed region, while some fluctuations exist in QSF.

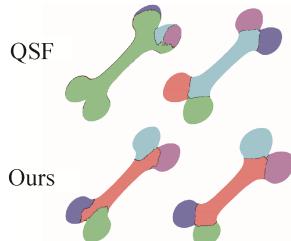


Fig. 9 Comparison of segmentation results with different initial methods. Left column: initial segmentation. Right column: final segmentation.

Table 2 Statistics of fitting error and running time.

Model	Error 0	Error 1	Time(s)
Bone	0.2483	0.0141	7.939
	0.0339	0.0140	1.467
Chess	0.0578	0.0087	2.214
	0.0325	0.0070	0.890
Sample	0.2354	0.0032	4.085
	0.0269	0.0030	0.908

Note: Error 0 and error 1 are fitting errors of initial segmentation and last segmentation, respectively. For each model, first row is QSF’s result^[13], and second row is our result.

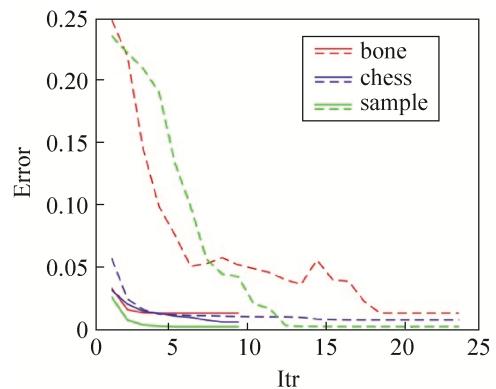


Fig. 10 Converge rate of our method and QSF^[13] in different models. The solid line stands for our method and dotted line stands for QSF.

Fig.11 shows the comparison of our method with the state-of-the-art methods: VSA^[1], QSF^[13], HFP^[23]. The result of VSA is worst, because it only uses plane for fitting, which is imprecise to fit non-plane parts of model. QSF method uses random initialization strategy, which is easy to fall into local optima. As shown in Fig.11(c), the right part of model is as a whole, and don't recognize detail of model base. HFP is a greedy hierarchical clustering approach, which is not appropriate to segment an object where the globally best fitting n parts need to be found. Compared with our result, HFP doesn't give a all right solution for this recognition problem.

Table 3 Statistics of models. $|F|$ is the number of triangles of input mesh; $|R|$ is the number of parts; last column is running times of segmentation process.

Model	$ F $ (K)	$ R $	Time(s)
Bone	30	5	1.467
Chess	8.5	10	0.890
Sample	26.7	21	0.908
Fandisk	13.8	21	1.497
Rolling sta.	100	23	2.010
Oil pump	100	78	15.43
Blade	390	37	5.120
Part4	143	93	14.324
Master cyl.	100	20	5.120
Grayloc	69.5	39	4.654
Carter	100	38	6.147
Dynamo	92.5	28	3.524

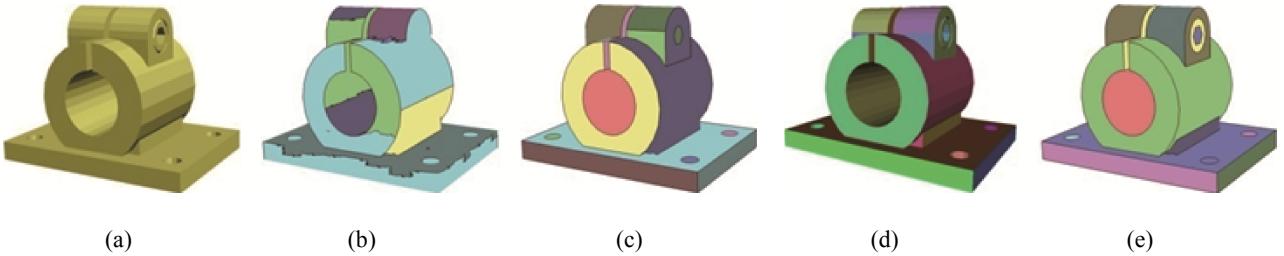


Fig. 11 Comparison with previous methods on the “dynamo” model. The number of parts is 28. For HFP(d), using plane, sphere and cylinder for fitting. (a) Input. (b) VSA. (c) QSF. (d) HFP. (e) Ours.

5.5 Application for mesh smoothing/denoising

A simple experiment have been performed to demonstrate the ability of our method to refine noise model, as shown in Fig.12. Given a clean 3D model (Fig.12(a)), we randomly select half of vertices and add Gaussian noise along with the normal of vertices to obtain a noise model (Fig.12(b)). The right segmentation result (Fig.12(c)) of this noise model can be obtained via proposed method. For each region, a best geometric primitive is assigned to it, and then all the vertices of noise model are projected into those fitted primitives. Specifically, for a vertex in inner

cluster, it is projected onto corresponding primitive; for a vertex in the boundary of two regions, it is projected into the boundary; for vertex in the junction of three regions, we first compute the intersection line of two regions, then compute the intersection point of this line and third regions. The Metro tool^[30] is applied to measure the symmetric Hausdoff distance between refined model and original clean model, as shown in Fig.12(d) and Fig.12(e). Clearly, the refined model with geometric constraint have smaller approximation error than that without geometric constraint.

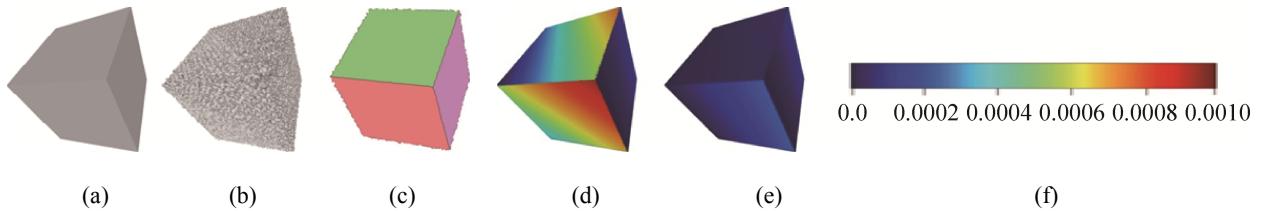


Fig. 12 Mesh smoothing/denoising. (a) Original model. (b) Noise model. (c) Segmentation result of noise model. (d) Fitting without geometric constraint. (e) Fitting with geometric constraint. (f) Colormap.

6 Conclusions

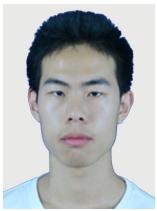
We have proposed a novel algorithm for recognizing geometric parts from scanned mechanical models. Based on the VSA framework, the RANSAC method is applied to obtain a good initial configuration, instead

of previous random initialization. Meanwhile, we use seed regions to guide re-grouping process, and this modification enhance the stability and robustness of the iterative optimization, especially for noise models. Furthermore, the geometric constraints are also introduced into surface fitting to force the

segmentation results to follow the global structure of the models. Various experimental results demonstrate the validity and robustness of our method, and it can be directly used for several geometric applications, such as mesh smoothing/denoising. In the future, we plan to add high-level constraints in our framework, e.g., symmetry. Moreover, we aim to extend this method to large scale building models and point clouds.

References

- [1] Huber D, Kapuria A, Donamukkala R, et al. Parts-based 3d object classification [C]//Proceedings of the 2004 IEEE Computer Society Conference on Computer Vision and Pattern Recognition. Washington: IEEE Computer Society, 2004: 82-89.
- [2] Chen X, Golovinskiy A, Funkhouser T. A benchmark for 3d mesh segmentation [J]. ACM Transactions on Graphics, 2009, 28(3): 1-12.
- [3] Park S B, Kim C S, Lee S U. Error resilient 3D mesh compression [J]. IEEE Transactions on Multimedia, 2006, 8(5): 885-895.
- [4] Cheng S C, Kuo C T, Wu D C. A novel 3D mesh compression using mesh segmentation with multiple principal plane analysis [J]. Pattern Recognition, 2010, 43(1): 267-279.
- [5] Lafarge F, Keriven R, Bredif M. Insertion of 3D-primitives in mesh-based representations: towards compact models preserving the details [J]. IEEE Transactions on Image Processing, 2010, 19(7): 1683-1694.
- [6] Sieger D, Gaulik S, Achenbach J, et al. Constrained space deformation techniques for design optimization [J]. Computer-Aided Design, 2016, 72: 40-51.
- [7] Cohen-Steiner D, Alliez P, Desbrun M. Variational shape approximation [J]. ACM Transactions on Graphics, 2004, 23(3): 905-914.
- [8] Marinov M, Kobbelt L. A Robust two-step procedure for quad-dominant remeshing [J]. Computer Graphics Forum, 2006, 25(3): 537-546.
- [9] Wu J, Kobbelt L. Structure recovery via hybrid variational surface approximation [J]. Computer Graphics Forum, 2005, 24(3): 277-284.
- [10] Simari P D, Singh K. Extraction and remeshing of ellipsoidal representations from mesh data [C]// Proceedings of Graphics Interface 2005, Canadian Human-Computer Communications Society. Waterloo: School of Computer Science. Victoria, British Columbia: University of Waterloo, 2005: 161-168.
- [11] Julius D, Kraevoy V, Sheer A. D-charts: quasi-developable mesh segmentation [J]. Computer Graphics Forum, 2005, 24(3): 981-90.
- [12] Yan D M, Liu Y, Wang W. Quadric surface extraction by variational shape approximation [C]//Proceedings of the 4th International Conference on Geometric Modeling and Processing. Berlin: Springer, 2006: 73-86.
- [13] Yan D M, Wang W, Liu Y, et al. Variational mesh segmentation via quadric surface fitting [J]. Computer-aided Design, 2012, 44(11): 1072-1082.
- [14] Li Y, Wu X, Chrysathou Y, et al. Globfit: consistently fitting primitives by discovering global relations [J]. ACM Transactions on Graphics, 2011, 30(4): 52.
- [15] Oesau S, Lafarge F, Alliez P. Planar shape detection and regularization in tandem [J]. Computer Graphics Forum, 2016, 35(1): 203-215.
- [16] Schnabel R, Wahl R, Klein R. Efficient ransac for point-cloud shape detection [J]. Computer Graphics Forum, 2007, 26(2): 214-226.
- [17] Shamir A. A survey on mesh segmentation techniques [J]. Computer Graphics Forum, 2008, 27(6): 1539-1556.
- [18] Theologou P, Pratikakis I, Theoharis T. A comprehensive overview of methodologies and performance evaluation frameworks in 3d mesh segmentation [J]. Computer Vision and Image Understanding, 2015, 135: 49-82.
- [19] Papaioannou G, Karabassi E A, Theoharis T. Segmentation and surface characterization of arbitrary 3d meshes for object reconstruction and recognition [C]//Pattern Recognition, 2000. Proceedings. 15th International Conference on. New York: IEEE Press, 2000: 734-737.
- [20] Lavoue G, Dupont F, Baskurt A. A new CAD mesh segmentation method, based on curvature tensor analysis [J]. Computer-Aided Design, 2005, 37(10): 975-987.
- [21] Jagannathan A, Miller E L. Three-dimensional surface mesh segmentation using curvedness-based region growing approach [J]. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2007, 29(12): 2195-2204.
- [22] Hwan K D, Dong Y I, Uk L S. Boundary-trimmed 3d triangular mesh segmentation based on iterative merging strategy [J]. Pattern Recognition, 2006, 39(5): 827-838.
- [23] Attene M, Falciadino B, Spagnuolo M. Hierarchical mesh segmentation based on tting primitives [J]. The Visual Computer, 2006, 22(3): 181-193.
- [24] Mangan A P, Whitaker R T. Partitioning 3d surface meshes using watershed segmentation [J]. IEEE Transactions on Visualization and Computer Graphics, 1999, 5(4): 308-321.
- [25] Katz S, Tal A. Hierarchical mesh decomposition using fuzzy clustering and cuts [J]. ACM Transactions on Graphics, 2003, 22(3): 954-961.
- [26] Lai Y K, Zhou Q Y, Hu S M, et al. Feature sensitive mesh segmentation [C]//Proceedings of the 2006 ACM Symposium on Solid and Physical Modeling, New York: Association for Computing Machinery, 2006: 17-25.
- [27] Lloyd S. Least squares quantization in pcm [J]. IEEE Transactions on Information Theory, 1982, 28(2): 129-137.
- [28] Guo J, Yan D M, Jia X, et al. Efficient maximal poisson-disk sampling and remeshing on surfaces [J]. Computers & Graphics, 2015, 46(C): 72-79.
- [29] Pauly M, Gross M, Kobbelt L P. Efficient simplification of point-sampled surfaces [C]//Proceedings of the Conference on Visualization '02. Washington: IEEE Computer Society, 2002: 163-170.
- [30] Cignoni P, Rocchini C, Scopigno R. Metro: measuring error on simplified surfaces [J]. Computer Graphics Forum, 1998, 17(2): 167-174.



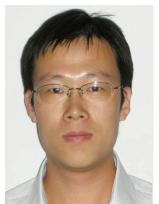
Quan Weize received his bachelor degree from Wuhan University of Technology in 2014, and he is currently a Ph.D. candidate in National Laboratory of Pattern Recognition (NLPR), Institute of Automation, Chinese Academy of Sciences. His research interests include 3D shape analysis and geometry processing.



Guo Jianwei is an assistant researcher in National Laboratory of Pattern Recognition (NLPR), Institute of Automation, Chinese Academy of Sciences (CASIA). He received his Ph.D. degree in computer science from CASIA in 2016, and bachelor degree from Shandong University in 2011. His research interests include computer graphics, geometry processing and 3D shape analysis.



Zhang Xiaopeng is a professor in National Laboratory of Pattern Recognition at Institute of Automation, Chinese Academic of Sciences (CAS). He received his Ph.D. degree in Computer Science from Institute of Software, CAS in 1999. He received the National Scientific and Technological Progress Prize (second class) in 2004. His main research interests include computer graphics and image processing.



Yan Dongming is an associate professor in the National Laboratory of Pattern Recognition (NLPR) at Institute of Automation, Chinese Academy of Sciences. He received his PhD from Hong Kong University in 2010, and his Master and Bachelor degrees both from Tsinghua University in 2005 and 2002, respectively. His research interests include computer graphics, geometric processing and visualization.