Computer-Aided Design 42 (2010) 1178–1188

Contents lists available at ScienceDirect

Computer-Aided Design

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

Feature suppression based CAD mesh model simplification

Shuming Gao *, Wei Zhao, Hongwei Lin, Fanqin Yang, Xiang Chen

State Key Lab of CAD&CG, Zhejiang University, Hangzhou, China

ARTICLE INFO

Article history:
Received 3 June 2009
Accepted 17 May 2010

Keywords:
Mesh simplification
Mesh segmentation
Feature recognition
Feature suppression
CAD/CAE

ABSTRACT

Dynamic simulation and high quality FEA mesh generation need the CAD mesh model to be simplified, that is, suppressing the detailed features on the mesh without any changes to the rest. However, the traditional mesh simplification methods for graphical models can not satisfy the requirements of CAD mesh simplification. In this paper, we develop a feature suppression based CAD mesh model simplification framework. First, the CAD mesh model is segmented by an improved watershed segmentation algorithm, constructing the region-level representation required by feature recognition. Second, the form features needing to be suppressed are extracted using a feature recognition method with user defined feature facility based on the region-level representation, establishing the feature-level representation. Third, every recognized feature is suppressed using the most suitable one of the three methods, i.e. planar Delaunay triangulation, Poisson equation based method, and the method for blend features, thus simplifying the CAD mesh model. Our method provides an effective way to make CAD mesh model simplification meet the requirements of engineering applications. Several experimental results are presented to show the superiority and effectivity of our approach.

1. Introduction

Triangular mesh models have been widely used in three-dimensional CAD systems for supporting engineering applications, such as dynamic simulation and finite element mesh generation. As CAD mesh models of complex products become more and more complicated, in order to improve the efficiency in simulating complex CAD mesh models dynamically, the CAD mesh model should be simplified in advance. Meanwhile, to make the simulation meet the requirements of engineering precision, the simplified CAD mesh models should retain large features of the original model. In other words, the CAD mesh model simplification should suppress the detailed features on the mesh without any changes to the rest. It differs from the mesh simplification for computer graphics applications.

As far as Finite Element Analysis (FEA) is concerned, the FEA mesh quality is critical to the FEA results. FEA computation on a good mesh with high quality can lead to results with high precision. However, it is a time-consuming task to generate a FEA mesh from CAD model. In fact, both the FEA mesh generation time and generated mesh quality mainly depend on the geometric shape of the CAD model. Generally speaking, the simpler the geometric shape of a CAD model, the less time the FEA mesh generation costs, and the better the quality of the generated mesh. Usually, the shape of a CAD model is made complicated by some small features on the model, which are inessential to FEA results, but prolong the FEA mesh generation time, and worsen the FEA mesh quality. In recent years, researchers started to explore generating FEA mesh directly based on CAD mesh model. Similarly, to shorten the mesh generation time and improve the mesh quality, the CAD mesh model is required to be simplified, that is, suppressing the detailed features on the mesh without any changes to the rest before FEA mesh generation.

Since the 1990's, mesh models have been extensively employed in computer graphics, and many research works on mesh simplification have been conducted [1–6] mainly for real-time rendering and fast transferring of complex mesh models. However, since the objects and purposes of mesh simplification for computer graphics and engineering applications are different, the mesh simplification methods for computer graphics are not suitable for simplifying CAD mesh models. First, the graphics mesh model normally reconstructed from 3D scanned data differs from the CAD mesh model generated from a B-rep model in CAD system greatly. Generally, the graphics mesh is dense, while the CAD mesh is very sparse. Especially, while graphics mesh simplification mainly considers the requirements of real-time rendering and fast transferring, CAD mesh simplification should retain the accuracy of major form, function and intent information of a product, which has been proven by the fact that practicing analysis normally simplifies CAD models by completely suppressing detailed form features while keeping other parts unchanged. Since the graphics mesh simplification methods simplify both detailed and major form features involved in a mesh model, they cannot retain the

* Corresponding author. Tel.: +86 571 88206681x514; fax: +86 571 88206680.
E-mail addresses: smgao@cad.zju.edu.cn (S. Gao), hwlin@zjucadcg.cn (H. Lin).

0010-4485/$ – see front matter © 2010 Elsevier Ltd. All rights reserved.
doi:10.1016/j.cad.2010.05.010

Fig. 1. Illustration of different simplification effects. (a) Original CAD mesh model; (b) Simplification result using the graphics mesh simplification method; (c) Simplification result by our method for CAD mesh simplification.

accuracy of the model's major form and thus cannot satisfy the requirements of engineering applications. As an example, Fig. 1(c) shows the simplified model of the original mesh model illustrated in Fig. 1(a), which is required by engineering applications, and Fig. 1(b) shows the simplified model generated by graphics mesh simplification methods. Obviously two simplified models are quite different.

To make the CAD mesh simplification meet the engineering requirements, we presented a feature suppression based CAD mesh simplification framework [7]. Given a CAD mesh model, it is first segmented into patches using an improved watershed method. Subsequently, features are recognized based on the topology of the segmented patches. Finally, these features are suppressed and the left holes are filled to generate the simplified CAD mesh model (Fig. 1(c)). In this paper, this algorithm is extended and intended to be able to effectively deal with complex features whose base faces could be any kind of surfaces. Specifically, our approach has the following contributions:

(1) An improved watershed algorithm that takes multi-descent strategy and performs iterative region merging is developed and used to achieve patch-type segmentation on CAD mesh models;
(2) A feature recognition algorithm for recognizing form features from CAD mesh models is proposed, which can recognize the pre-defined features and user-defined features that are interactively defined by the users during the model simplification process;
(3) Poisson equation based feature suppression and blend feature suppression methods are developed to conduct feature suppression for the form feature with curved base region and the blend features, respectively, which effectively guarantees the simplification quality.

The paper is organized as follows. Section 2 surveys related works. Section 3 introduces some concepts and the overview of the algorithm. The improved watershed algorithm is developed in Section 4. In Sections 5 and 6, we presents the feature recognition and suppression methods, respectively. Section 7 illustrates some results. Finally, this paper is concluded in Section 8.

2. Related works

As stated above, there are only a few works on feature recognition and simplification of CAD mesh models. The first work on CAD mesh simplification based on form feature suppression was proposed by Jang et al. [8] in 2006. Their approach extracts form features involved in a mesh model using loop-based feature detector and simplify the model through feature suppression. Restrained by their loop-based feature detection algorithm, the approach can only deal with the very simple feature whose base face is a single planar face. Recently, Sunil and Pande [9] reported the design and implementation of a system for automatic recognition of features from freeform surface CAD models of sheet metal parts represented in STL format. However, in their work, the method for mesh segmentation is heuristic and the method for feature recognition cannot recognize user-defined features. Kim et al. [10] proposed a B-rep model simplification method to speed up rendering and smooth user interaction with three operators: wrap-around, smooth-out and thinning. These operators work well on their applications but not suitable to engineering applications. Their operators will largely change the shape and volume of the part and generate non-manifold models.

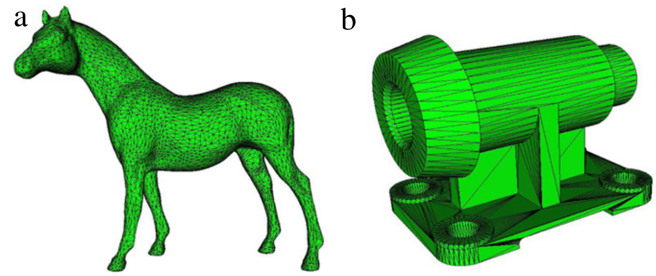

Fig. 2. Computer graphics mesh (a) and CAD mesh (b).

Besides the graphics mesh simplification methods and the works above on the CAD mesh model, the works on segmentation, feature recognition and feature suppression are also relevant and thus they are briefly introduced below.

2.1. Mesh segmentation

In general, mesh segmentation methods can be classified into two types: patch-type segmentation and part-type segmentation. The former is mainly used for texture mapping, chart building, simplification and re-meshing, while the latter usually divides a mesh into meaningful parts without restricting the part topology. Several approaches to automatically segment mesh into meaningful components have been proposed. The watershed-based scheme put forward by Mangan and Whitaker favors partitioning a mesh model along high curvature regions [11]. Kim et al. [12] presented a mesh partitioning method using iterative merging strategy. A mean shift-based segmentation method is proposed by Yamauchiy et al. [13]. Lee et al. [14] presented an intelligent manual scissoring tool for mesh using the minima rule and part salience. Ji [15] proposed an easy mesh cutting tool to segment a mesh based on an improved region growing algorithm using a feature sensitive metric. Katz et al. [16] proposed a hierarchical mesh segmentation algorithm which is based on new methods for prominent feature point and core extraction. Yu-Kun Lai et al. [17] presented a top-down hierarchical mesh segmentation algorithm based on isotropic re-meshing. Attene et al. [18] described a hierarchical face clustering algorithm for triangle meshes based on fitting primitives belonging to an arbitrary set. After defining the term 'regular object', Varady et al. [19] presented a non-iterative algorithm for direct segmentation, where well-known techniques from computer vision are combined with new procedures for processing point and normal vector data. A comprehensive discussion of mesh segmentation methods was achieved by Attene [20].

In general, the most existing mesh segmentation methods are computer graphics oriented and work better on triangulated computer graphics models with dense mesh. Their emphasis is how to divide a streamlined triangular mesh (see Fig. 2(a)) into meaningful pieces. But the computer graphics mesh models are much different from CAD mesh models, which are very sparse, highly non-uniform, not streamlined and have many hard edges (see Fig. 2(b)). Therefore, these segmentation methods for computer graphics models cannot be applied to the CAD models directly. To make mesh segmentation really meet the requirements of engineering applications, there is still a long way to go.

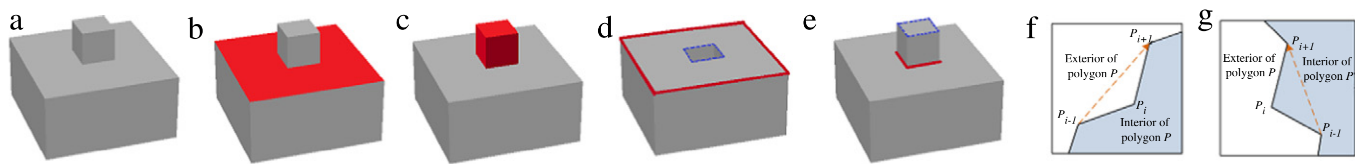

Fig. 3. Illustration of basic concepts. (a) Regions; (b) Base region (darker); (c) Feature regions (darker); (d) Inner loop (dash line) and outer loop (solid line); (e) Convex edges (dash line) and concave edges (solid line); (f) Concave vertex (P_i); (g) Convex vertex (P_i).

2.2. Feature recognition

Automated form feature recognition has been an active research topic in solid modeling area for many years and many methods have been proposed. These methods can be divided into three categories: graph-based algorithms [21–24], volumetric decomposition techniques [25–28] and hint-based geometric reasoning [29–31].

Graph-based feature recognition algorithms translate the B-rep of a part model into a graph where its nodes represent faces and arcs represent edges with some attributes such as the convexity and concavity of edges included, and then recognize all the form features by graph decomposition and sub-graph matching.

In volume decomposition approaches, a part is decomposed into a set of intermediate volumes using convex hull decomposition or cell decomposition first, then all the form features are produced by combining the intermediate volumes in a certain way.

Hint-based approaches assert that any feature will leave a trace in the part boundary which provides a hint for the potential existence of a feature. So in these approaches features are extracted by geometric reasoning based on the feature hints.

For a detailed feature recognition survey, please refer to Ref. [32]. As mentioned in Ref. [32], although much progress has been made in feature recognition techniques, the complete problem is still far from being resolved.

2.3. Feature suppression

The real feature suppression works mainly focus on the suppression of detailed features like blend features. Li [33] presented a suppression algorithm for blend features generated by rolling-ball techniques with a fixed radius. An approach to the suppression of blend features with a variational radius was proposed by Zhu [34]. Joshi [35] put forward a feature suppression method which can deal with blend features between two NURBS surfaces. Cui [36] proposed an effective algorithm for suppressing vertex blend, edge blend and mixed blend features. Lee et al. [37] suppressed detailed features through constructing a feature volume. Venkataraman et al. [38,39] presented methods to reconstruct the features from a set of faces of a solid model, and remove the face set of the features from the model.

The key technique issue of suppressing the features except blend features on mesh model is hole-filling. Many hole-filling approaches have been proposed during the last decade and they can be classified into two categories: voxel-based [40–43] and triangle-based [44–47]. However, due to the complexity and diversity of the issue, existing hole-filling methods still have certain deficiencies in robustness, efficiency and precision, especially for complex models with highly curved holes.

3. Preliminaries

3.1. Basic concepts

(1) *CAD mesh model*: A triangular mesh model generated by triangulating B-rep model in CAD system.
(2) *Region*: A set of connected triangles of a CAD mesh model, normally corresponding to a face of the B-rep (Fig. 3(a)).
(3) *Loop*: An ordered and interconnected edge set which forms the closed boundary of a region. There are two types of loops: outer loop and inner loop. The outer boundary of a region forms the outer loop, while the boundary of an inner hole or protrusion determines the inner loop (see red lines in Fig. 3(d)).
(4) *Concave/convex edge*: An edge is concave/convex if the angle between its two adjacent triangles is smaller/larger than π (Fig. 3(e)).
(5) *Concave/convex vertex*: A 2D vertex P_i is concave/convex if a left/right turn is made at P_i while going from P_{i-1} to P_{i+1}, where the interior of the 2D polygon P is to the right (see Fig. 3(f) and (g)).
(6) *Concave/convex connection*: The connection between two adjacent regions is concave/convex if they are connected by concave/convex edges (Fig. 3(e)).
(7) *Base region*: The region which has at least one inner loop or one concave vertex in its outer loop, on which at least one feature is attached (Fig. 3(b)).
(8) *Feature region*: The regions involved in a feature (Fig. 3(c)).
(9) *Feature boundary*: The boundary between all the regions of the feature and all the adjacent regions of the feature.

The basic concepts presented above are illustrated in Fig. 3.

3.2. Approach overview

In this paper, we propose a feature suppression based framework to CAD mesh model simplification. Firstly, a CAD mesh model is segmented into regions using the improved watershed algorithm, and the region-level representation of the model is set up; secondly, all the form features are recognized based on the region-level representation using the graph-based feature recognition method with a user-defined feature facility; finally, each recognized feature is suppressed by a suitable one of three feature suppression methods. Additionally, a hierarchical data structure consisting of the triangle-level, region-level and feature-level of mesh models is designed and used in our approach to effectively support the feature suppression based CAD mesh simplification.

The flowchart of our approach is illustrated in Fig. 4.

4. Mesh segmentation

As aforementioned, though the graphics mesh segmentation methods are not suitable to segment the CAD mesh models directly, mesh segmentation technique in computer graphics field has been developed for a long time, and there are lots of classical methods for mesh segmentation. Therefore, to segment a CAD mesh model, a feasible way is to improve the classical mesh segmentation method for graphics model and make it suitable for the CAD mesh model. In this section we improve the classical watershed method to partition the CAD mesh model into a set of regions with face granularity. And then, the region-level representation of the mesh model can be established, which consists of all the regions as well as relevant information among

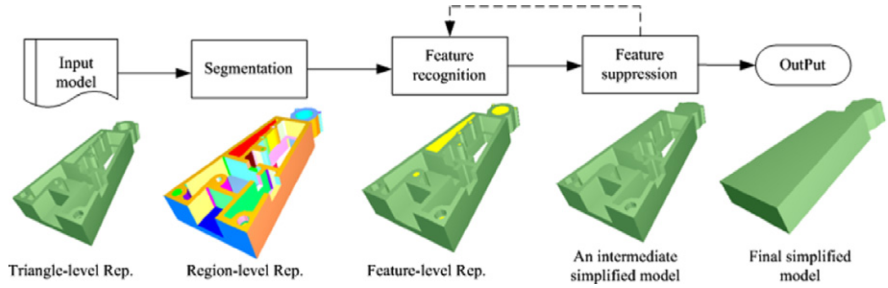

Fig. 4. Flowchart and illustration of our approach.

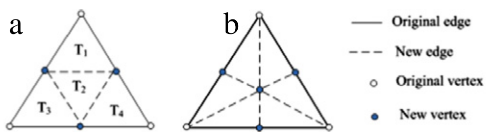

Fig. 5. Adding new vertices in a triangle. (a) Manner 1; (b) Manner 2.

them, e.g. connection properties between two adjacent regions (convex or concave), region topology, etc.

Different from the mesh models that are reconstructed from 3D scanned data or used in the computer graphics area, CAD mesh models generated by triangulating B-rep in CAD systems have the following characteristics (Fig. 2(b)): (1) the triangle quality is usually not good, i.e. there are slender and sparse triangles in the CAD mesh model; (2) there are hard boundaries, which are determined by the nature of CAD parts. These characteristics make traditional segmentation methods for graphics models unsuitable for partitioning CAD mesh models. For example, the original watershed algorithm cannot find a vertex with a local minimum curvature as the catchment basin for sparse CAD mesh. Thus, the classical watershed algorithm for graphical models must be improved greatly to fit CAD mesh segmentation. The improvements include, adding new vertices to refine the mesh, taking a multi-descent strategy during the segmentation and finally performing iterative region merging.

For clarity, the steps of the improved watershed algorithm are listed as follows:

Step 1: Refine the mesh by adding new vertices to the original mesh model;
Step 2: Segment the refined mesh model using a multi-descent strategy;
Step 3: Update the region representation generated based on the refined mesh model;
Step 4: Generate the final regions by iterative region merging.

4.1. Mesh refinement by adding new vertices

Since CAD mesh models usually contain many large and sparse triangles, there are not enough vertices to calculate the curvature and define a catchment basin. Therefore, the traditional watershed algorithm cannot segment CAD mesh models with high quality. This problem can be solved by adding new vertices into the original mesh properly, that is, refining the original mesh. To hold the features of the CAD mesh model, the refinement of the original mesh cannot change its geometry. To this end, two vertex insertion manners are presented in the following, based on the feature edge concept introduced by Razdan [48] (An edge is called a feature edge if the normals of its two adjacent triangles form an angle greater than a given threshold.):

Manner 1: If not all edges of a triangle are feature edges, three new vertices are added at the mids of three edges of the triangle, creating four sub-triangles (see Fig. 5(a));

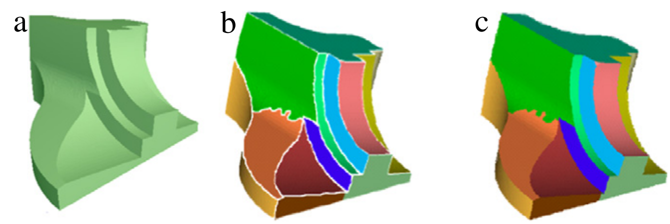

Fig. 6. Illustration of hard boundary problem. (a) A mesh model; (b) Segmentation result without hard boundary, segmented by the traditional watershed method; (c) Segmentation result with hard boundary, segmented by our improved watershed method.

Manner 2: If the three edges of a triangle are all feature edges, a new vertex is added at the center of the triangle and three new vertices are added at the mids of three edges, generating six sub-triangles (see Fig. 5(b)).

In this way, the features of the original CAD mesh model are remained precisely. Moreover, compared with Razdan's method [48], our approach can maintain the topology of the mesh model much better.

4.2. Segmentation by multi-descent strategy

One major limitation of traditional watershed algorithm is that the segmentation result has no hard boundary [48]. That is, there remain undetermined strips between patches segmented (see blank triangles in Fig. 6(b)). Razdan et al. proposed a method to determine the boundary in the strips based on triangle decomposition in Ref. [49]. However, since they have to modify the original mesh, the approach is not suitable for engineering applications.

In fact, it is mainly because the descent strategy in the traditional watershed method is unidirectional and only one region label can be assigned to each vertex, that the hard boundary cannot be generated. Take the case in Fig. 7 as an example, since the vertex D in Fig. 7(a) is a *ridge* one, it should be assigned not only the region label R_1 of C but the region label R_2 of E. However, with the unidirectional descent strategy, the traditional watershed algorithm only flows downhill on the height function towards the lowest value point C, and thus only the region label R_1 of C is assigned to D. Therefore, the edge DE is undetermined, and we do not know which region it belongs to (Fig. 7(b)).

To overcome the above limitations of the traditional watershed algorithm, in this paper, we develop a new descent strategy, i.e. the multi-descent strategy, to segment the CAD mesh models. Differing from the unidirectional descent strategy, the multi-descent strategy enables the watershed algorithm to flow downhill from a vertex towards all its neighbor vertices which have lower height function values than that of the vertex. In this way, the region label can be assigned to each vertex according to the following rules:

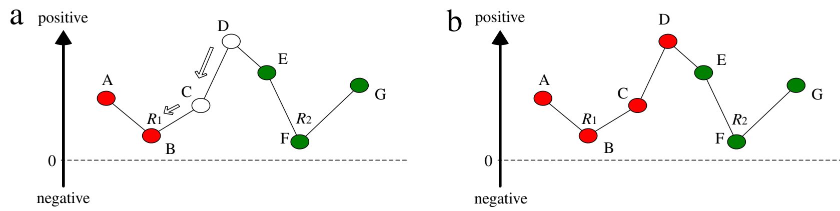

Fig. 7. Unidirectional descent strategy. (a) Descent path of vertex D; (b) Descent result.

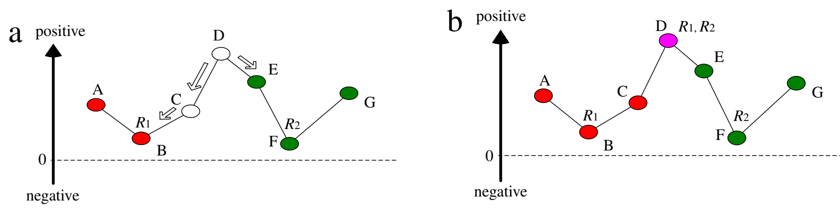

Fig. 8. Multi-descent strategy. (a) Multi-descent path of vertex D; (b) Descent result.

(1) At the vertex on a *ridge*, the descent along multiple directions can get to different minimum value vertices, so it is assigned multiple region labels.
(2) At the vertex not on a *ridge*, the descent along multiple directions can arrive at only one minimum value vertex, so it is assigned only one region label.

Therefore, the vertex with multiple region labels is the vertex on the boundary of the regions segmented. In this way, the hard boundary is determined (Fig. 6(c)). For clarity, we illustrate the multi-descent strategy in Fig. 8, where vertex D gets both region labels R_1 and R_2, and thus it is considered as a boundary vertex.

4.3. Region representation update

After the segmentation with a multi-descent strategy is fulfilled, the refined mesh model is decomposed into a number of regions each of which consist of a set of connected triangles of the refined mesh model. Now we need to update each region representation by replacing the triangles of the refined mesh model with these of the original mesh model, so as to make the segmentation result be represented by the original triangles. The triangle replacement is performed as follows.

On one hand, for the triangle of the original mesh model in which some new vertices are added by manner 1 (Fig. 5(a)), the sub-triangles are replaced according to the following rules:

(1) If three or more sub-triangles of the original triangle belong to the same region, then the original triangle is added to this region and all its four sub-triangles are deleted;
(2) Otherwise, the original triangle is added to the region to which T_2 (Fig. 5(a)) belongs and all its four sub-triangles are deleted.

On the other hand, for the triangle of the original mesh model in which some new vertices are added by manner 2 (Fig. 5(b)), since all its six sub-triangles must belong to the same region after the segmentation because the new vertex on the center is the vertex with local minimum curvature, the original triangle is added to the region to which its six sub-triangles belong and all six sub-triangles are deleted.

4.4. Generation of final segmentation result by iterative region merging

While adding new vertices is very helpful for generating the hard boundaries, it often leads to oversegmentation, especially on the blend areas. Therefore, region merging is required to generate the final segmentation result.

To do so, two queues are created: One stores the *candidate regions* (*CR*), and the other stores the *adjacent candidate region pairs* (*ACRP*), which are defined as follows, as well as the *valid adjacent region pair*.

Valid adjacent region pair: An adjacent region pair $\langle R_1, R_2\rangle$ is valid if all dihedral angles between two adjacent boundary triangles are smaller than an angle threshold δ_{angle} and the area ratio α between R_1 and R_2 satisfies $\alpha \in [1/\delta_{\text{area}}, \delta_{\text{area}}]$, where $\delta_{\text{area}} \geq 1$.

Candidate region: A region is a candidate region if it belongs to at least one valid adjacent region pair.

Adjacent candidate region pair: An adjacent region pair $\langle R_1, R_2\rangle$ is a candidate one if both R_1 and R_2 are candidate regions.

With *CR* and *ACRP* defined above, we can delineate the region merging algorithm, which consists of the following four steps:

Step 1: Initialize the *CR* and *ACRP* queues by determining all the candidate regions and adjacent candidate region pairs of the segmented mesh model according to the above definitions;
Step 2: Get the front pair from the *ACRP* queue, and merge these two regions if the new merging region is still a candidate region;
Step 3: Update the region representation in the *CR* and *ACRP* queues by replacing merged regions with new region;
Step 4: Repeat Step 2 to Step 3 till the *CR* or *ACRP* queue is empty or a given iteration time is reached.

Fig. 9 shows two examples of the final blend region generated by the region merging, where the first column shows the original mesh models, the second column demonstrates the segmentation results based on the refined mesh models with newly added vertices, and the third column illustrates the final blend regions after region merging.

4.5. Discussion on the improved watershed segmentation method

The improved watershed segmentation method works well on CAD models with hard edges. Meanwhile, it still needs to be improved in the following two aspects: (1) Capability of handling the model including complex blending features and the model containing two adjacent curved surfaces which are triangulated with similar triangles, e.g. their triangle shapes and areas are similar; (2) Effective selection of the parameters in the improved watershed segmentation algorithm that directly determine the segmentation quality.

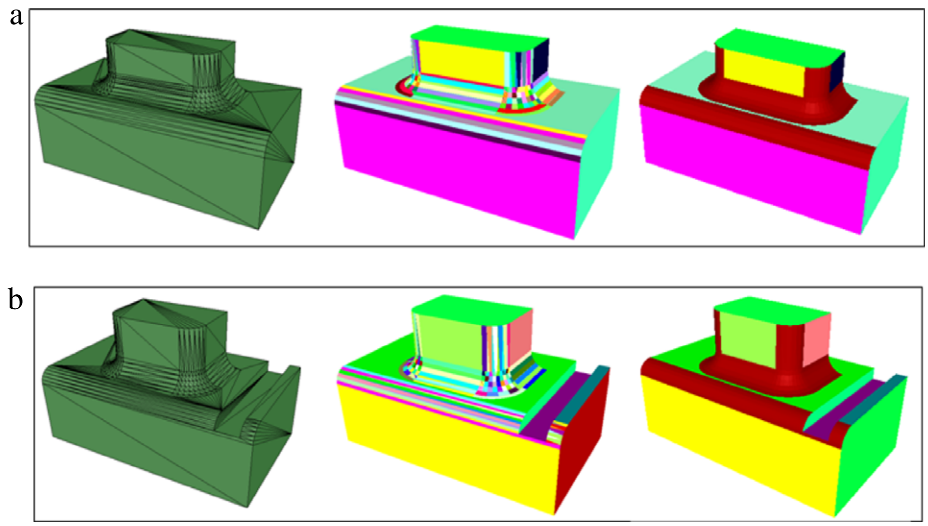

Fig. 9. Two examples of the final blend region generation by iterative region merging. (a) Example 1; (b) Example 2.

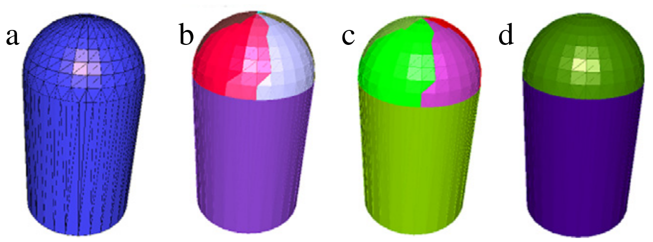

Fig. 10. Sensitivity of the parameter *curvature*. (a) The original model; (b) Curvature = 0.01; (c) Curvature = 0.05; (d) Curvature = 0.07.

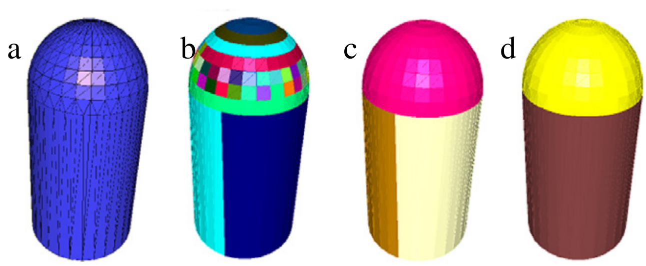

Fig. 11. Sensitivity of the parameter *angle*. (a) The original model; (b) Angle = $(1/18)\pi$; (c) Angle = $(1/9)\pi$; (d) Angle = $(1/5)\pi$.

Currently, for complicated mesh models, to generate desirable segmentation results, the parameters in the improved watershed segmentation algorithm need to be selected experimentally and experientially according to the requirements and the actual triangulation result which CAD software exports. The major parameters as well as their sensitivity are illustrated as follows.

Curvature: Curvature is the key parameter for segmentation. It is the decisive factor to whole segmentation. Generally speaking, the higher the curvature is, fewer parts the segmented model has. Taking Fig. 10 as an example, with the curvature threshold varying from 0.01 to 0.07, the segmented model has less and less parts (angle and area ratio are ignored):

Angle: Angle controls the size of region. The parameter merges two adjacent regions when their dihedral is less than the angle threshold. The larger the angle is, bigger the region is. Fig. 11 shows the segmentation trend when the angle is getting larger (we fix curvature threshold = 0.01, area ratio = 2).

Area ratio: The area ratio is designed for splitting two adjacent regions whose triangles' areas are different. As we know, CAD software triangulates the model by surface. The triangles lying on the same surface have the same shape. The area ratio splits the model using this characteristic. Fig. 12 shows the segmented results with a varying area ratio (curvature = 0.01, angle = $(1/5)\pi$).

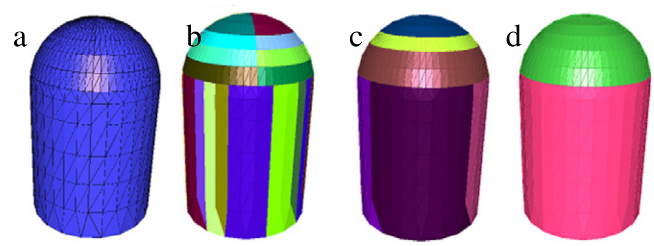

Fig. 12. Sensitivity of the parameter *area ratio*. (a) The original model; (b) Area ratio = 1.1; (c) Area ratio = 1.2; (d) Area ratio = 1.3.

5. Feature recognition

To suppress the features and simplify the CAD mesh model, these features should be recognized in advance. With the region-level representation of the mesh model, the graph-based method [50] is employed and improved to recognize features. Specifically, two major modifications are made to the traditional graph-based feature recognition method in our work, that is:

(1) Since the features interested in this work are mainly detailed features and are usually attached to one or more base faces, we decompose the face-edge graph of the whole model based on the base faces rather than only convex faces;
(2) Because only feature boundary information is required by feature suppression, we define rib features completely by their face-edge graph without feature parameters and constraints, which makes the definition and representation of rib features not only simple but also unified, and facilitates the implementation of the user defined feature recognition.

In general, there has been a pre-defined feature library. However, due to the diversity of the features that need to suppressed, it is impossible to pre-define all of them in a feature library. In this work, we allow users to interactively define the features that they want to suppress during the process of mesh model simplification. The user-defined features can also be put into the feature library and used as pre-defined features afterward.

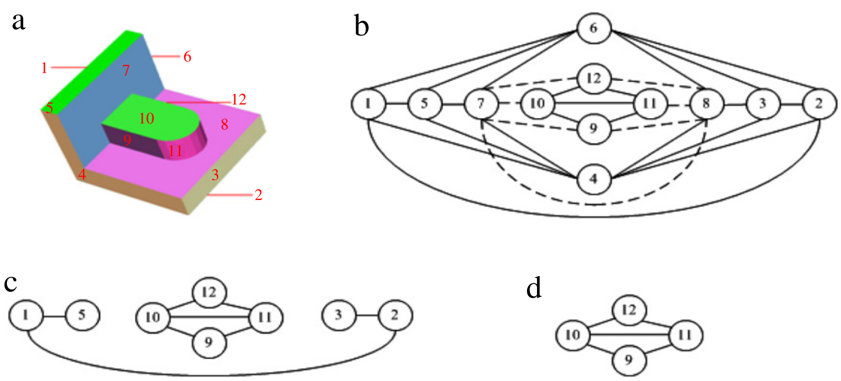

Fig. 13. Process of feature recognition based on RAG. (a) Mesh model with region information; (b) RAG of the mesh model; (c) Sub-graphs obtained by graph decomposition; (d) Feature graph candidate.

With the user-defined and pre-defined feature library in hand, the features in the CAD mesh can be recognized according to the following steps (Fig. 13).

Set up region adjacency graph (*RAG*): Based on the region-level representation of the mesh model, the region adjacency graph (*RAG*) of the model can be set up, in which each node refers to a region and each arc represents the adjacency relationship between the two regions including convex/concave attribute. An example is shown in Fig. 13(b) where solid lines refer to convex adjacency relationship, and dashed lines stand for the concave adjacency relationship.

Find out all base regions: All the base regions can be easily determined according to the base region definition given in Section 3.1. Specifically, for each region, whether it has inner loop (s) or concave vertex (vertices) is checked. If so, it is taken as a base region since both the inner loop and concave edge indicate that there is some form feature attached on this region. In Fig. 13(a), regions 4, 6, 7 and 8 are determined as base regions.

Decompose the region adjacency graph (*RAG*): In order to reduce the space of sub-graph matching that is the most time-consuming step of the graph-based feature recognition, we decompose the *RAG* of the mesh model by deleting all the base regions from it. It is because a base region will not be involved in any form feature that needs to be suppressed. After all the base regions are deleted, the *RAG* is decomposed into a set of sub-graphs, each of which might correspond to a form feature needing to be suppressed. For the *RAG* shown in Fig. 13(b), after the graph decomposition by deleting base regions 4, 6, 7 and 8, two sub-graphs {1, 2, 3, 5} and {9, 10, 11, 12} are obtained, as depicted in Fig. 13(c).

Determine feature graph candidates: For each sub-graph obtained in the previous step, we further check whether it may correspond to a real feature needing to be suppressed, i.e. whether it is a *feature graph candidate*, by checking if the regions of the sub-graph are adjacent with its base regions on inner loop edges or concave edges. If so, the sub-graph may correspond to a real feature needing to be suppressed and is taken as a feature graph candidate. Moreover, after a feature graph candidate is determined, it is checked whether all the regions of the feature graph candidate have a common region with a concave adjacency with them. If so, this common region is restored and added into the feature graph candidate. In Fig. 13(c), sub-graph {9, 10, 11, 12} is determined as the feature graph candidate, as shown in Fig. 13(d), while sub-graph {1, 2, 3, 5} is excluded.

Sub-graph matching: In the last step, each feature graph candidate is compared with the *RAG* of every pre-defined feature in the feature library. If it matches any one, a feature is recognized. For the example shown in Fig. 13, the feature graph candidate {9, 10, 11, 12} is finally recognized as a feature.

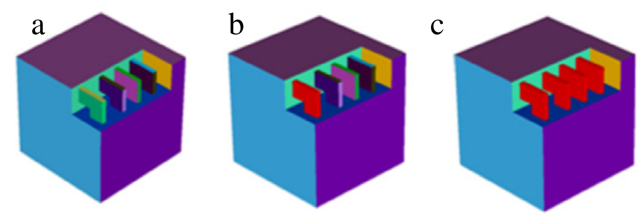

Fig. 14. Define and recognize user-defined features. (a) The original model; (b) Defining of a new feature (red); (c) Recognized new features (red). (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

Fig. 14 demonstrates an example for recognizing a user-defined feature. Fig. 14(a) shows the original model segmented. In Fig. 14(b), user defines a new feature in red, and the features are all recognized in Fig. 14(c).

6. Feature suppression

After all the features that need to be suppressed are recognized, they can be suppressed to simplify the CAD mesh model. Specifically, feature suppression includes two steps: (1) deleting every recognized feature, i.e. deleting all the triangles involved in every feature from the mesh model; and (2) filling the holes formed due to the feature deletion. Since feature deletion is very simple, we focus on hole-filling. In this section, according to the geometric shape of the suppressed feature, we develop three hole-filling methods, each for a certain type hole.

6.1. Planar Delaunay triangulation method

If all the base regions of the feature to be suppressed are planar, the hole formed by removing the feature will be filled using the planar Delaunay triangulation method [49,51,52]. There are two cases:

Case 1: If the feature has only one planar base region, indicating that the hole lies completely on a single planar region, then it is filled by directly performing planar Delaunay triangulation with the boundary of the hole as input and adding all the new triangles generated into the mesh model.

Case 2: If the feature has more than one planar base regions, meaning the hole is not a planar hole, e.g. the rib feature shown in Fig. 15(a), we first decompose the boundary of the hole into a number of planar loops, each of which lies on a base region of the feature, and then fill every generated loop by planar Delaunay triangulation with the loop as input and adding all the new triangles generated into the mesh model. For example, after the

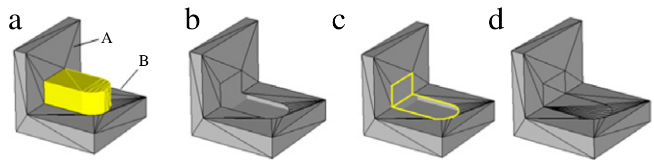

Fig. 15. Suppression of rib feature with two base regions. (a) The feature-level representation of the mesh model (A and B refer to the two base regions of the rib feature); (b) The mesh model with rib feature deleted; (c) Two loops generated by hole decomposition; (d) The final feature suppression result.

Fig. 16. An example of feature suppression based on the Poisson equation. (a) A mesh model with a feature whose base region is curved (red); (b) Effect of suppression based on planar triangulation; (c) Effect of suppression based on the Poisson equation.

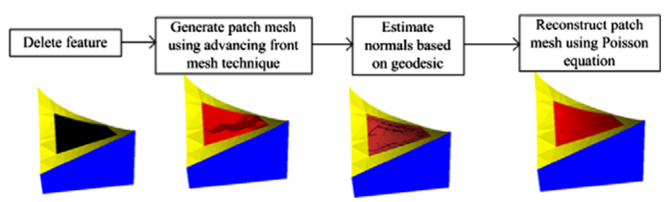

Fig. 17. The flowchart and illustration of feature suppression based on the Poisson equation.

rib feature in Fig. 15(a) is removed, the hole is first decomposed into two loops as shown in Fig. 15(c), and each loop is filled by planar Delaunay triangulation. The final feature suppression result is shown in Fig. 15(d).

6.2. Poisson equation method

If the base region of a feature is curved, the effect of the planar triangulation method is not desirable (Fig. 16(a) and (b)). In order to effectively fill the hole on the curved region so that the new triangles can be compatible with the base region, we proposed a hole-filling algorithm based on normal estimation and Poisson equation [53]. The flowchart and illustration of the algorithm is shown in Fig. 17. Fig. 16(c) shows the suppression effect of the feature in Fig. 16(a) using this algorithm, which is obviously better than that shown in Fig. 16(b). For the detailed algorithm, please refer to Ref. [53].

6.3. Blend feature suppression

Due to the speciality of blend features, the two feature suppression methods presented above cannot achieve the desirable suppression effect (see Fig. 19) for them. Thus, we develop the following algorithm for suppressing blend features according to their characteristic. This suppression method can suppress blending features generated by the ball-rolling algorithm.

Firstly, since the max principle curvature lines of a blend region always flow from one of its adjacent regions to another (Fig. 18(a)), all the vertices on the blend region can be classified into a number of clusters by tracing seed vertices, each of which correspond to a max principal curvature line.

Secondly, the junction vertices, that is, the common vertices between the blend region and its adjacent regions, are determined

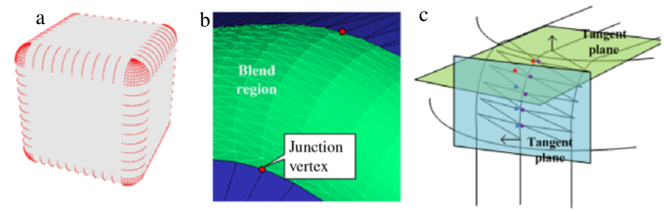

Fig. 18. Illustration of some notations used in blend feature suppression. (a) Maximal principle curvature flow lines; (b) Two junction vertices on a curvature flow line; (c) The tangent planes at two junction vertices and the projection of the vertices on them.

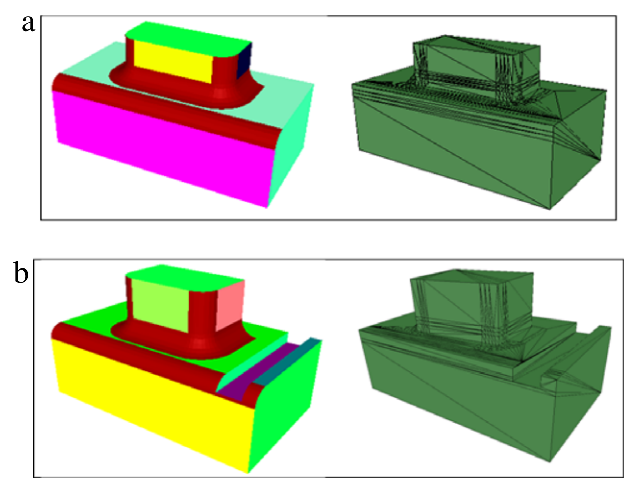

Fig. 19. Two examples of blend feature suppression. (a) Example 1; (b) Example 2.

during tracing of the max curvature lines. As shown in Fig. 18(b), each vertex cluster has two junction vertices.

Thirdly, for each cluster, we construct two tangent planes at their two junction vertices (Fig. 18(c)), each of which takes a junction vertex as its root point and the normal of the junction vertex on the adjacent region as its normal. Then every vertex of the cluster is projected onto the tangent plane closest to it along its normal, as shown in Fig. 18(c).

Finally, check whether all the triangles in the blending feature have their three vertices lying on the same tangent plane. If there is a vertex lying on a tangent plane which is different from that of the other two vertices, then project the vertex onto the intersection line between two tangent planes.

The above algorithm simplifies the shape of a blend region by changing the coordinates of every vertex of the blend region without making any modification to the topology of the blend region. Fig. 19 shows two examples of suppressing blend regions using the algorithm.

After planar and blend feature suppression, we need a procedure to reduce the number of coplanar triangles. In this work, we choose the method of Garland and Heckbert [5] for this task. By selecting an error threshold close to zero, we can achieve the desired effect.

7. Implementation and results

The proposed approach has been implemented with a VC++ 2003 environment. And various CAD mesh models have been used to test our approach. In the following, several representative examples are illustrated.

Fig. 20 shows the Chasis model and its simplification results. Fig. 20(b) is the segmentation result with curvature threshold $T = 0.01$, angle threshold $\delta_{\text{angle}} = 0.2\pi$, and area threshold $\delta_{\text{area}} = 6$; Fig. 20(c) shows the recognized pre-defined features; Fig. 20(d)

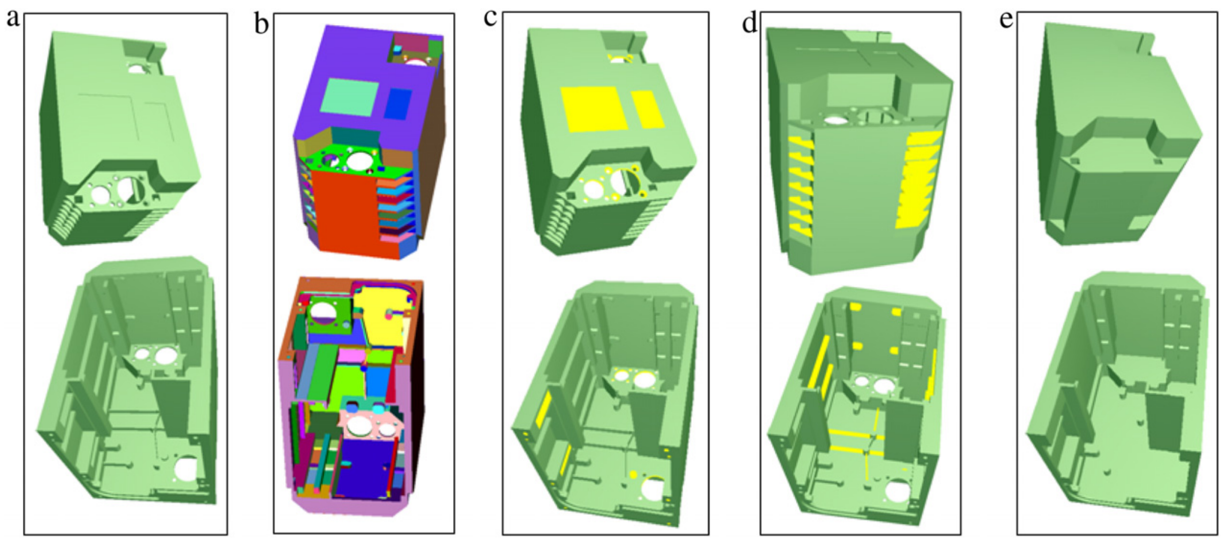

Fig. 20. The simplification of Chasis model. (a) Original mesh model; (b) Region-level representation; (c) Recognized pre-defined features; (d) Recognized user-defined features; (e) Final simplification result.

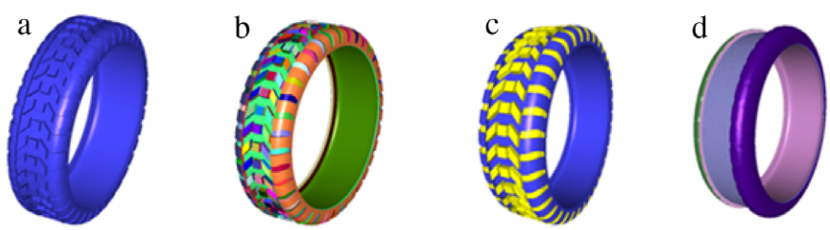

Fig. 21. The simplification of tire model. (a) Original mesh model; (b) Segmentation result; (c) All recognized features; (d) Final simplification result.

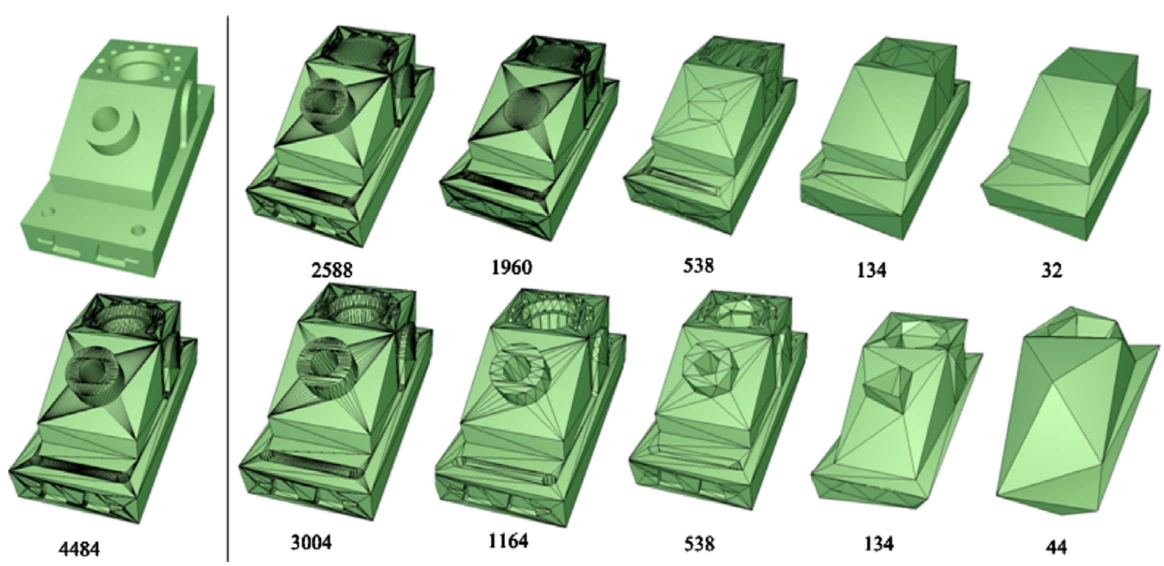

Fig. 22. The comparison between our method and Garland and Heckbert's method.

depicts the recognized user-defined features; and Fig. 20(e) is the final simplified result. Fig. 21 shows the Trye model and its simplification result. From this example, it can be seen that our approach has the capacity of suppressing the form features whose base regions are curved.

Furthermore, a comparison is demonstrated in Fig. 22 between our approach and one of the traditional simplification methods (i.e. the method of Garland and Heckbert [5]) with respect to the simplification effect on CAD mesh models. In Fig. 22, the left column shows a CAD mesh model with 4484 triangles (one is in shading and another is wireframe), and the other columns show the simplified models (the simplified models on the top are the simplification results of our approach and those on the bottom are the results of Garland and Heckbert's method). Obviously, the final simplification effects of two methods are very different and our approach can satisfy the requirements of engineering applications better. In addition, it can be seen that our approach can also generate the multi-resolution simplified models as Garland and Heckbert's method does.

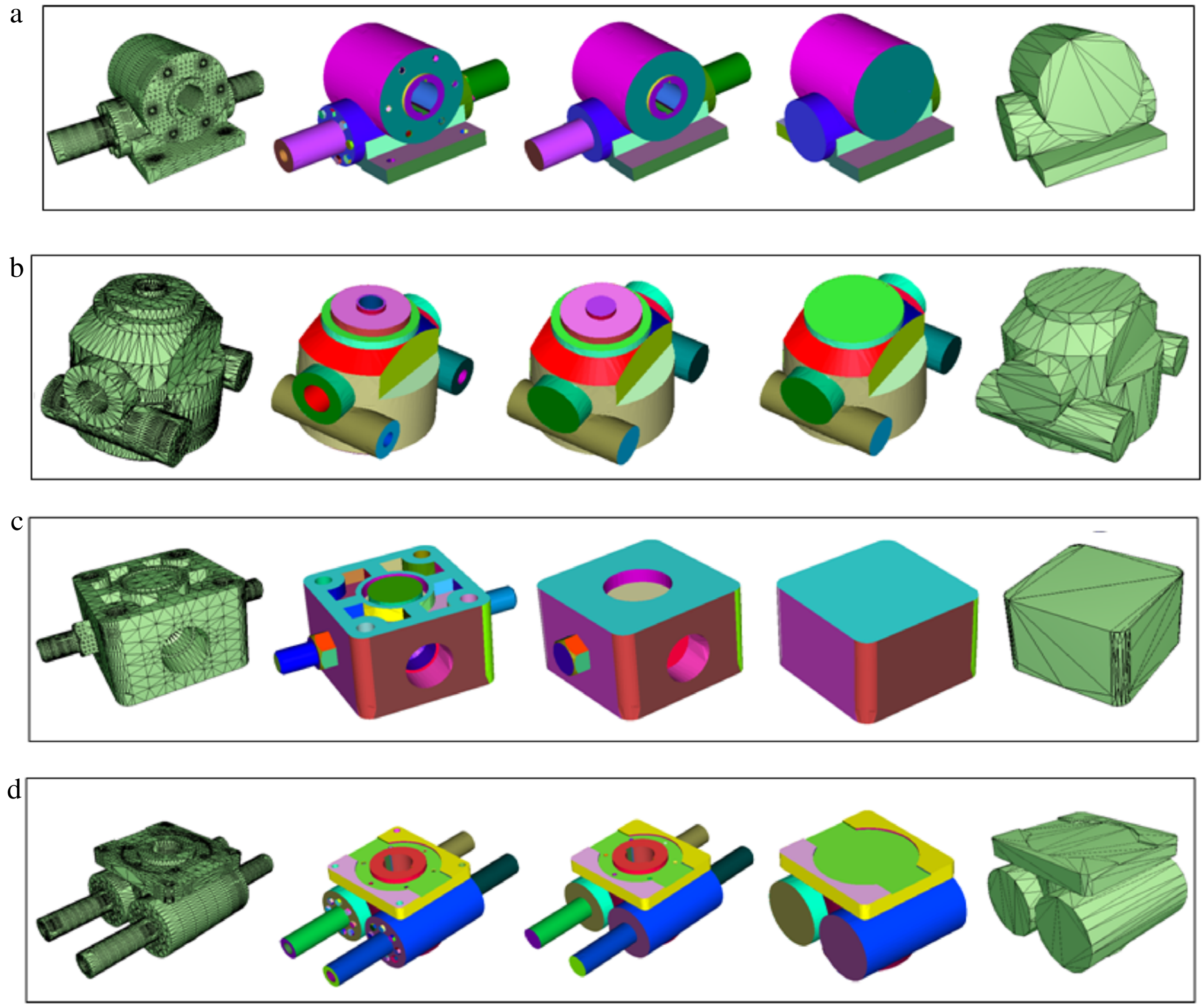

Fig. 23. Simplification of four parts.

In Fig. 23, more examples are shown, each of which consists of five models. The first and second are the initial mesh model and the mesh model with segmentation results, and the last three are the multi-resolution simplified models.

8. Conclusions and future work

In this paper, we present a framework to CAD mesh model simplification based on feature suppression. Firstly, a CAD mesh model is segmented into patches using an improved watershed method. Secondly, pre-defined features or user-defined features are recognized based on the topology of the segmented patches. Finally, these recognized features are suppressed and the left holes are filled to generate the simplified CAD mesh model. The framework is intended to be able to effectively deal with complex CAD mesh models. Compared with traditional simplification algorithms, our approach can meet the requirements of engineering applications better for it is capable of removing unnecessary form features from the mesh model without any changes to the rest part.

There are five aspects for future work: First, we want to design an efficient method to select the parameters in the segmentation. Second, a robust segmentation algorithm is required, which is dedicated to the CAD mesh segmentation. Third, we need to design an easy way to control the level of simplification. Fourth, for better filling of the hole, we should develop a method to subdivide the loops constituting the boundary of the hole into pieces, each for a base region. Finally, a physical based feature suppression method will be designed in the future.

Acknowledgements

The authors are very grateful for the financial support from NSF of China (Nos. 60736019, 60970150 and 60933008), 863 High Tech. Plan of China (No. 2008AA04Z115), and Zhejiang Provincial Natural Science Foundation of China (Nos. Z107497, Y1090416).

References

[1] Rossignac J, Borrel P. Multi-resolution 3D approximation for rendering complex scenes. In: Falcidieno B, Kunii T, editors. Modeling in computer graphics: methods and applications. Proceedings of conference, Berlin: Springer-Verlag; 1993. p. 455–65.
[2] Turk G. Re-tiling polygonal surfaces. Computer Graphics 1992;26(2):55–64.
[3] Scheder WJ, Zarge JA, Lorensen WE. Decimation of triangles meshes. Computer Graphics 1992;26(2):65–70.
[4] Hoppe H, Derose T, Mcdonal J, Stuetzle W. Mesh optimization. In: SIGGRAPH'93 proceedings. ACM SIGGRAPH. p. 19–26.
[5] Garland M, Heckbert PS. Surface simplification using quadric error metrics. In: Proceedings of SIGGRAPH'97. ACM SIGGRAPH. 1997. p. 209–16.

[6] Lindstrom P, Turk G. Fast and memory efficient polygonal simplification. In: Proceedings of IEEE visualization'98. 1998. p. 279–86.
[7] Gao SM, Zhao W, Yang FQ, Chen X. Feature suppression based CAD mesh model simplification. In: 10th IEEE international conference on shape modeling and applications. 2008.
[8] Jang J, Wonka P, Ribarsky W, Shaw CD. Punctuated simplification of man-made objects. The Visual Computer 2006;22(2):136–45.
[9] Sunil VB, Pande SS. Automatic recognition of features from freeform surface CAD models. Computer-Aided Design 2008;40(4):502–17.
[10] Kim S, Lee K, Hong T, Kim M, Jung M, Song Y. An integrated approach to realize multi-resolution of B-rep model. In: Proceedings of ACM solid and physical modeling conference. 2005. p. 153–62.
[11] Mangan A, Whitaker R. Partitioning 3D surface meshes using watershed segmentation. IEEE Transactions on Visualization and Computer Graphics 1999;5(4):308–21.
[12] Kim DH, Yun ID, Lee SU. Boundary-trimmed 3D triangular mesh segmentation based on iterative merging strategy. Pattern Recognition 2006;39(5):827–38.
[13] Yamauchiy H, Lee S, Le Y, Ohtake Y, Belyaevy A, Seidel HP. Feature sensitive mesh segmentation with mean shift. In: Processing of shape modeling international. 2005. p. 236–43.
[14] Lee Y, Lee S, Shamir A, Cohen D, Seidel HP. Mesh scissoring with minima rule and part salience. Computer-Aided Geometric Design 2005;22(5):444–65.
[15] Ji Z, Liu L, Chen Z, Wang G. Easy mesh cutting. Computer Graphics Forum 2006; 25(3):283–91.
[16] Katz S, Leifman G, Tal A. Mesh segmentation using feature point and core extraction. The Visual Computer 2005;21(8):649–58.
[17] Lai YK, Zhou QY, Hu SM, Martin RR. Feature sensitive mesh segmentation. In: Proceedings of the 2006 ACM symposium on solid and physical modeling. 2006. p. 17–25.
[18] Attene M, Falcidieno B, Spagnuolo M. Hierarchical mesh segmentation based on fitting primitives. The Visual Computer 2006;22(3):181–93.
[19] Varady T, Benko P, Kos G. Reverse engineering regular objects: simple segmentation and surface fitting procedures. International Journal of Shape Modeling 1998;4(3):127–41.
[20] Attene M, Katz S, Mortara M, Patané G, Spagnuolo M, Tal A. Mesh segmentation: a comparative study. In: Proceedings shape modeling and applications. 2006. p. 14–25.
[21] Joshi S, Chang TC. Graph based heuristics for recognition of machined features from a 3-D solid model. Computer-Aided Design 1988;20(2):58–66.
[22] Marefat M, Kashyap RL. Geometric reasoning for recognition of 3-D object features. IEEE Transactions on Pattern Analysis and Machine Intelligence 1990; 12(10):949–65.
[23] Ji Q, Marefat M. Bayesian approach for extracting and identifying features. Computer-Aided Design 1995;27(6):435–54.
[24] Trika SN, Kashyap RL. Geometric reasoning for extraction of manufacturing features in iso-oriented polyhedrons. IEEE Transactions on Pattern Analysis and Machine Intelligence 1994;16(11):1087–100.
[25] Kyprianou LK. Shape classification in computer aided design. Ph.D. dissertation. Cambridge (UK): Christ College. Univ. Cambridge; July 1980.
[26] Woo T. Feature extraction by volume decomposition. In: Proceedings conference CAD/CAM technology in mechanical engineering. 1982.
[27] Kim Y. Convex decomposition and solid geometric modeling. Ph.D. dissertation. Stanford Univ.; 1990.
[28] Kim Y. Recognition of form features using convex decomposition. Computer-Aided Design 1992;24(9):461–76.
[29] Vandenbrande JH, Requicha AAG. Spatial reasoning for the automatic recognition of machinable features in solid models. IEEE Transactions on Pattern Analysis and Machine Intelligence 1993;15(12):1–17.
[30] Vandenbrande JH, Requicha AAG. Geometric computation for the recognition of spatially interacting machinable features. In: Shah JJ, Mäntylä M, Nau DS, editors. Advances in feature based manufacturing. Amsterdam (The Netherlands): Elsevier Science B.V.; 1994. p. 83–106.
[31] Han J, Requicha AAG. Hint generation and completion for feature recognition. In: Proc. int. symp. automotive technology and automation. 1996. p. 89–96.
[32] Shah JJ, Anderson David, Kim Yong Se, Joshi Sanjay. A discourse on geometric feature recognition from CAD models. Journal of Computing and Information Science in Engineering 2001;1(1):41–51.
[33] Li B, Liu J. Detail feature recognition and decomposition in solid model. Computer-Aided Design 2002;34(5):405–14.
[34] Zhu H, Menq CH. *B*-rep model simplification by automatic fillet/round suppressing for efficient automatic feature recognition. Computer-Aided Design 2002;34(2):109–23.
[35] Joshi N, Dutta D. Feature simplification techniques for freeform surface models. Journal of Computing and Information Science in Engineering 2003; 3(3):177–86.
[36] Cui X.F., Gao S.M., Zhou G.P.. An effective algorithm for recognizing and suppressing blend features, in Computer-Aided Design and Applications, May 24–28. Pattaya Beach, Thailand. 2004.
[37] Lee KY, Armstrong CG, Price MA, Lamont JH. A small feature suppression/unsuppression system for preparing *B*-rep models for analysis. In: Proceedings of the 2005 ACM symposium on solid and physical modeling. 2005. p. 113–24.
[38] Venkataraman S, Sohoni M. Reconstruction of feature volumes and feature suppression. In: Proceedings of the seventh ACM symposium on solid modeling and applications. 2002. p. 60–71.
[39] Venkataraman S, Sohoni M, Rajadhyaksha R. Removal of blends from boundary representation models. In: Proceedings of the seventh ACM symposium on solid modeling and applications. 2002. p. 83–94.
[40] Curless B, Levoy M. A volumetric method for building complex model from range image. In: Proceedings of SIGGRAPH. 1996. p. 303–12.
[41] Davis J, Marschner SR, Garr M, Levoy M. Filing holes in complex surface using volumetric diffusion. In: Proceedings of 3D data processing visualization and transmission. 2002. p. 428–33.
[42] Ju T. Robust repair of polygonal models. In: Proceedings of SIGGRAPH. 2004. p. 888–95.
[43] Joshua P, Szymon R. Atomic volumes for mesh completion. In: Proceedings eurographics symposium on geometry processing. 2005. p. 33–41.
[44] Carr J, Beatson R, Cherrie J, Mitchell T, Fright W, McCallum B. Reconstruction and representation of 3D objects with radial basis functions. In: Proceedings of SIGGRAPH. 2001. p. 67–76.
[45] Chen CY, Cheng KY, Liao HYM. A sharpness dependent approach to 3D polygon mesh hole filling. In: Proceedings of eurographic. Trinity College, Blackwell Press; 2005. p. 13–6.
[46] Jun Y. A piecewise hole-filling algorithm in reverse engineering. Computer-Aided Design 2005;22(8):263–70.
[47] Liepa P. Filling hole in meshes. In: Proceedings eurographics symposium on geometric processing. 2003. p. 200–7.
[48] Razdan A, Bae MS. A hybrid approach to feature segmentation of triangles meshes. Computer-Aided Design 2003;35(9):783–9.
[49] Dey TK. Delaunay triangulation in three dimensions with finite precision arithmetic. Computer Aided Geometric Design 1992;9(6):457–70.
[50] Gao SM, Shah JJ. Automatic recognition of interacting machining features based on minimal condition sub-graph. Computer-Aided Design 1998;30(9): 727–39.
[51] Girod B, Greiner G, Niemann H. Principles of 3D image analysis and synthesis. Boston: Kluwer; 2000.
[52] O'Rourke J. Computational geometry in C. New York: Cambridge University Press; 1999.
[53] Zhao W, Gao SM, Lin HW. A robust hole-filling algorithm for triangular mesh. The Visual Computer 2007;23(12):987–97.