Computer-Aided Design 44 (2012) 1072–1082

Contents lists available at SciVerse ScienceDirect

Computer-Aided Design

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

# Variational mesh segmentation via quadric surface fitting

Dong-Ming Yan [a,b,*], Wenping Wang [a], Yang Liu [c], Zhouwang Yang [d]

[a] Department of Computer Science, University of Hong Kong, Hong Kong
[b] GMSV Center, KAUST, Thuwal 23955-6900, Saudi Arabia
[c] Microsoft Research Asia, Haidian District, Beijing, 100800, PR China
[d] University of Science and Technology of China, Hefei 230026, PR China

ARTICLE INFO

*Article history:*
Received 1 November 2011
Accepted 24 April 2012

*Keywords:*
Mesh segmentation
Quadric surface fitting
Error metric

ABSTRACT

We present a new variational method for mesh segmentation by fitting quadric surfaces. Each component of the resulting segmentation is represented by a general quadric surface (including plane as a special case). A novel energy function is defined to evaluate the quality of the segmentation, which combines both $L^2$ and $L^{2,1}$ metrics from a triangle to a quadric surface. The Lloyd iteration is used to minimize the energy function, which repeatedly interleaves between mesh partition and quadric surface fitting. We also integrate feature-based and simplification-based techniques in the segmentation framework, which greatly improve the performance. The advantages of our algorithm are demonstrated by comparing with the state-of-the-art methods.

© 2012 Elsevier Ltd. All rights reserved.

## 1. Introduction

Compact and faithful representation of 3D objects is crucial for various applications in computer graphics, computer vision and CAD/CAM community. Nowadays freeform surfaces are popularly represented by triangular meshes, which can easily be obtained with high accuracy and complexity thanks to the rapid development of 3D digital data acquisition devices. But such meshes are too raw to be directly used in the subsequent process due to the lack of a high-level representation, even with preprocessing such as denoising, simplification or remeshing.

Mesh segmentation is one of the most effective ways to compute a high-level shape representation. E.g., in reverse engineering, the input scanned data are first segmented into simple surface patches, and the intersection curves of adjacent patches are then computed to form the boundary representation (*B*-Rep) [1–3]. Other applications, such as texture atlas generation [4,5], remeshing [6,7] and compression [8] can also benefit from mesh segmentation.

Most previous shape segmentation approaches are based on local clustering or boundary detection techniques. However, their greedy nature tends to make the segmentation unsatisfactory. Cohen-Steiner et al. [6] have introduced the *Variational Shape Approximation* (VSA) framework, which segments the input mesh surface into planar patches. The quality of the segmentation is measured by the sum of the normal derivation of each cluster from its corresponding planar proxy. Lloyd iteration [9] is used to minimize the energy function. Wu and Kobblet extend the VSA framework by introducing spheres and circular cylinders as basic fitting primitives [10]. Other primitives, such as ellipsoidal surfaces [11], developable patches [12] are also studied.

Quadric surfaces are preferred for surface approximation in many applications because they have low algebraic degree and their shapes are easy to control [13]. In this paper, we shall present a new algorithm to segment input mesh surfaces (especially scanned or tessellated industrial CAD models) into non-overlapping patches, each patch approximated by a general quadric surface; see Fig. 1 for an example.

This paper is an extension of our previous work [14]. Instead of using only $L^2$ or $L^{2,1}$ metrics separately, we introduce a novel error function that measures both the geometric distance and the normal derivation between a patch of a mesh surface and a fitting quadric surface. We will demonstrate the advantages of the new error function over the previous work which uses only the $L^2$ metric in Section 6. The efficiency of our algorithm is further improved by using the feature information and the mesh simplification techniques. The main contributions of this paper include:

- a variational mesh segmentation framework based on fitting general quadrics (including planes as a special case);
- a new error function for fitting quadric surface from original mesh triangles, where both $L^2$ and $L^{2,1}$ distance are considered;
- a new method for smoothing irregular boundary curves between adjacent segmented regions using a graph-cut method;
- enhancements of the variational segmentation framework, including feature-based and simplification-based segmentation.

* Corresponding author at: GMSV Center, KAUST, Thuwal 23955-6900, Saudi Arabia. Tel.: +966 2 8080469.
*E-mail addresses:* yandongming@gmail.com (D.-M. Yan), wenping@cs.hku.hk (W. Wang), yangliu@microsoft.com (Y. Liu), zhouwang.yang@gmail.com (Z. Yang).

0010-4485/$ – see front matter © 2012 Elsevier Ltd. All rights reserved.
doi:10.1016/j.cad.2012.04.005

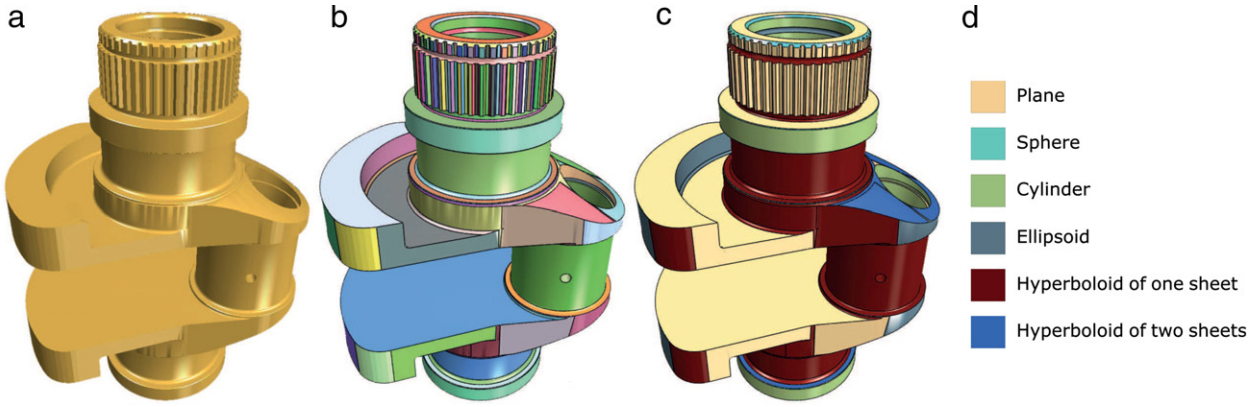


**Fig. 1.** Quadric surface segmentation of a crank model. (a) The mesh model with 100 K faces. (b) Segmentation result with 666 regions. The color of each segmented patch is randomly generated for visualization purpose. (c) Fitting result, different colors indicate different kind of quadric surfaces in (d). (d) Color index for different kinds of quadric surfaces. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

### 1.1. Related work

Mesh segmentation has been studied extensively in the past years. The aim of mesh segmentation is to partition the input mesh into "meaningful" components. The definition of "meaningful" can totally be different according to different applications. In the following we shall briefly discuss the existing work related to our approach.

#### 1.1.1. Mesh segmentation

Traditional mesh segmentation algorithms are based on the local property of surfaces. Surface elements with similar properties are grouped together to form larger patches [15]. This kind of approaches can be regarded as *local clustering* based methods, or *greedy approaches*. Compared with traditional algorithms, some recent *variational approaches* show powerful approximation ability and better segmentation quality.

*Greedy approaches. Region growing* is the most popular method in the literature for surface segmentation [1,16–18]. A set of seed points are first selected and for each seed grows a region until all the surface elements are assigned to a region. Local surface properties, such as principle curvatures, are always used as criteria for growing regions with the similar attribute. Region growing based methods always stop at boundaries with high curvature, e.g., a flat ellipsoid, which will be over segmented by such approaches.

*Hierarchical clustering* based methods merge the pair of regions from bottom to top hierarchically [19–23]. At the beginning, each face of the mesh is assigned as a single region. In each step, a pair of adjacent regions with least merging error is merged to form a new region. The algorithm is repeated until some stopping criteria is met. This kind of approaches have problems in blending regions between two smooth surfaces, which may be merged in the early stage.

*Hierarchical decomposition* based methods, also known as *mesh splitting*, segment mesh surfaces into meaningful components in a top-down manner. Many region splitting algorithms are based on the minima rule and part salience theory [24,25]. The feature curves on mesh surfaces are first detected and used as part of boundaries of final segmentation. Various meaningful metrics are defined for different applications [26–30]. This approach, tends to segment surface at concave regions. Some meaningful parts may be over segmented by this approach.

Region growing, hierarchical clustering and hierarchical decomposition are regarded as greedy approaches, because once the segmentation is done for a triangle element, it will not be changed in the later process. However, mesh segmentation can be treated as an energy minimization problem, where an energy function is defined and optimized for the segmentation. Because of its optimization nature, this method is often referred as *variational method*. The planarity and developability of surface regions are usually used as error metrics to define energy functions [31,6,12,32–35].

*Variational approaches.* Cohen-Steiner et al. [6] propose a new shape approximation algorithm by clustering face normal of the mesh, where consistent energy minimization is applied to drive down the approximation error. To keep the connectivity of each region, a distortion minimization flooding algorithm is developed. This method is efficient but only planar surfaces are used as fitting primitives, which tends to produce too many planar polygons for segmentation purpose. This work is extended in several ways by introducing higher order or special type of surface elements [11, 10,14]. In [11], ellipsoidal surface is used as the only type of primitive to approximate a given mesh by minimizing a combined energy function. The segmentation boundaries are smoothed by a constrained relaxation of the boundary vertices. To reduce the number of surface elements, Wu and Kobbelt [10] extend [6]'s work by introducing sphere, circular cylinder and rolling ball patch as basic primitives. The number of elements is reduced a lot but the type of basic primitives are still too restrictive to represent both CAD and free-form objects. Yan et al. [36] apply the variational segmentation framework for segmenting the laser scanned tree data into cylindrical components and then reconstruct the branch models of trees.

#### 1.1.2. Surface fitting

*Surface fitting* is a key step in many mesh segmentation algorithms. A detailed survey of surface fitting techniques is out of the scope of this paper. We shall focus on low-degree algebraic surface fitting approaches, specifically, quadric surface fitting.

Since there is no closed-form for computing foot point on quadric surfaces, direct fitting, which minimizes the Euclidean distance, is a non-linear optimization problem [37,38], which is too inefficient in practice. Given an implicit surface $f(\mathbf{x}) = 0$, the most well-known fitting method is to use algebraic distance $f(\mathbf{x})$ to approximate geometric distance, but this approximation is too biased even in simple cases [39,40]. Taubin approximates the Euclidean distance from a point to a quadric surface by a first order approximation $\frac{f(\mathbf{x})}{|\nabla f(\mathbf{x})|}$ [41]. By using this first order distance, the fitting problem can then be solved as a generalized eigenvector problem by a further approximation. Instead of considering only the geometric distance, many approaches introduce the normal deviation in the fitting algorithm [42–45]. Kanai et al. present a quadric fitting algorithm which combines both algebraic distance and gradient [46]. Instead of fitting only discrete sampled

point data, they integrate the error function over the triangular meshes. The proposed fitting algorithm is then used to construct a hierarchical implicit surface structure. Since only algebraic distance is used, and the gradient vector is not normalized, the fitting results behave bad in high curvature region (see Fig. 11). In this paper, we propose a new error function, which is a hybrid of a first order approximation of the geometric distance and the normal difference between the polygon face and the normalized gradient of the fitting surface. The new error function gives better fitting results (see Section 6).

### 1.2. Outline

The remainder of this paper is organized as follows. Section 2 derives the problem formulation of variational shape segmentation. A new error function for quadric surface fitting is introduced in Section 3, and the variational segmentation framework is introduced in Section 4. Section 5 presents two acceleration techniques to improve the efficiency of the segmentation algorithm. We present experimental results in Section 6 and draw our conclusion in Section 7.

## 2. Problem formulation

Let $\mathcal{M}$ be a finite set of triangles $\{t_j\}_{j=1}^m$ that constitute a connected mesh surface. A partition of $\mathcal{M}$ is denoted by $\mathcal{R} = \{\mathcal{R}_i\}_{i=1}^n$, where $\bigcup_{i=1}^n \mathcal{R}_i = \mathcal{M}$ and $\mathcal{R}_i \bigcap \mathcal{R}_j = \emptyset$ for any $i \neq j$. All the triangles of a subset $\mathcal{R}_i$ form a connected component. Each subset $\mathcal{R}_i$ is also called a *region*, or a *cluster*, which is fitted by a best fitting geometric proxy, denoted by $\mathcal{P}_i = \{s_i, f_i(\mathbf{x})\}$, where $s_i \in \mathcal{R}_i$ is the seed triangle of the proxy $\mathcal{P}_i$ and $f_i(\mathbf{x}) = 0$ is a general quadric surface, which can also be a plane. The objective function of this partition, for a fixed $n > 0$, is defined by

$$E(\mathcal{R}) = \sum_{i=1}^{n} E(\mathcal{R}_i, \mathcal{P}_i) = \sum_{i=1}^{n} \sum_{t_j \in \mathcal{R}_i} E(t_j, f_i), \tag{1}$$

where $E(t_j, f_i)$ is a metric measuring the cost of the triangle $t_j$ with respect to the fitting surface $f_i(\mathbf{x}) = 0$. The optimal partition $\mathcal{R}$ is a minimizer of Eq. (1). The metric $E(t_j, f_i)$ is application dependent and should be defined so that the optimal partition satisfies the requirement of that application.

The above objective function Eq. (1) can be minimized by Lloyd's algorithm [6]. Given an initial partition, two alternative steps, i.e., surface fitting and mesh partition, are performed to minimize the same energy function consistently. In the surface fitting step, Eq. (1) is minimized by fitting a quadric proxy to each region $\mathcal{R}_i$. In the partition step, Eq. (1) is minimized again by reassigning each triangle to its 'nearest' proxy to form a new partition.

## 3. Metric for quadric surfaces

The implicit equation of a quadric surface $f(\mathbf{x}) = 0$, where $\mathbf{x} = [x, y, z]^T \in \mathbb{R}^3$ is a 3D point, can be written as

$$f(\mathbf{x}) = \mathbf{C}^T \cdot \mathbf{F}, \tag{2}$$

where $\mathbf{C} = [c_0, c_1, \ldots, c_9]^T$ is the vector of unknown coefficients of the quadric surface and $\mathbf{F} = [1, x, y, z, xy, xz, yz, x^2, y^2, z^2]^T$. Both of them are ten dimensional vectors.

### 3.1. Triangle-quadric error metric

We take both $L^2$ and $L^{2,1}$ distance between a triangle $t$ and a quadric surface $f(\mathbf{x})$ into account. The hybrid distance is defined by

$$E(t, f) = E_d(t, f) + \omega E_n(t, f), \tag{3}$$

where $E_d$ measures the squared Euclidean distance ($L^2$) and $E_n$ measures the normal deviation ($L^{2,1}$) from $t$ to $f$. It is known that the exact distance between a point and a quadric surface can be computed by solving a 6th-degree univariate equation. Since we want to integrate the square distance over triangles instead of summing the distances of discrete samplings, it is difficult to use this exact computation which will be too inefficient. By balancing both the efficiency and accuracy, we use a first order approximation [41] instead of the exact distance. The hybrid distance between a triangle and a quadric surface is approximated by the following formula respectively:

$$E_d(t, f) = \int_t \frac{f(\mathbf{x})^2}{|\nabla f(\mathbf{x})|^2} \cdot d\sigma, \tag{4}$$

$$E_n(t, f) = \int_t \left( \frac{\nabla f(\mathbf{x})}{|\nabla f(\mathbf{x})|} - \mathbf{n}_t \right)^2 \cdot d\sigma,$$

where $\mathbf{n}_t$ is the unit normal vector of $t$.

### 3.2. Quadric surface fitting

A straight forward way to minimize the energy function $E(\mathcal{R}_i, \mathcal{P}_i)$ for each region requires to solve a non-linear least square optimization problem, which is quite time consuming. In this paper, we propose an alternative method to solve this problem efficiently in two steps.

#### 3.2.1. Initial surface fitting

Given a region $\mathcal{R}_i$, we first compute an initial quadric surface $f_i^0(\mathbf{x})$ using the algorithm presented in [14], which only takes the $L^2$ component into account. The objective function is defined by

$$\begin{aligned} E'(\mathcal{R}_i, \mathcal{P}_i) &= \sum_{t_j \in \mathcal{R}_i} E_d(t_j, f_i^0) \\ &\approx \frac{\sum_{t_j \in \mathcal{R}_i} \int_{t_j} f_i^0(\mathbf{x})^2 d\sigma}{\sum_{t_j \in \mathcal{R}_i} \int_{t_j} |\nabla f_i^0(\mathbf{x})|^2 d\sigma} \\ &= \frac{\mathbf{C}_0^T M_t \mathbf{C}_0}{\mathbf{C}_0^T N_t \mathbf{C}_0}, \end{aligned}$$

where $\mathbf{C}_0^T$ is the unknown coefficient vector of $f_i^0$, and $M_t, N_t$ are coefficient matrices. Hence, the minimization of $E'$ is reduced to computing the eigenvector of $M_t - \lambda N_t$ associated with the minimum eigenvalue [47].

#### 3.2.2. Least-square fitting

The initial fitting step results in a good guess of the unknown quadric surface. In the second step, we compute the gradient $\nabla f_i^0(\mathbf{c}_{t_j})$ for each triangle $t_j \in \mathcal{R}_i$ at the barycenter $\mathbf{c}_{t_j}$ of $t_j$. We use this one point approximation of the gradient for each triangle by substituting $|\nabla f_i^0(\mathbf{c}_{t_j})|$ into Eq. (4). The energy function of each region $\mathcal{R}_i$ is approximated as below:

$$\begin{aligned} E(\mathcal{R}_i, \mathcal{P}_i) \approx & \sum_{t_j \in \mathcal{R}_i} \int_{t_j} \frac{f(\mathbf{x})^2}{|\nabla f_i^0(\mathbf{c}_{t_j})|^2} \cdot d\sigma \\ & + \omega \sum_{t_j \in \mathcal{R}_i} \int_{t_j} \left( \frac{\nabla f(\mathbf{x})}{|\nabla f_i^0(\mathbf{c}_{t_j})|} - \mathbf{n}_t \right)^2 \cdot d\sigma. \end{aligned}$$

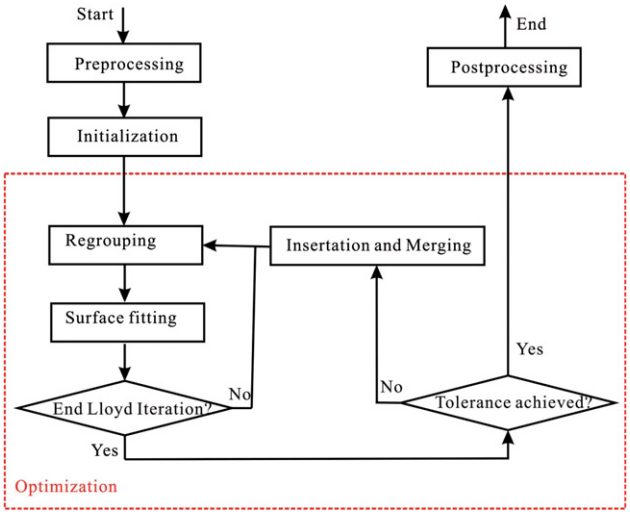


**Fig. 2.** Flowchart of the variational segmentation framework.

With this simplification, the optimization of the energy function for each region $\mathcal{R}_i$ is reduced to a linear least square problem, which is similar to that proposed in [46]. We use Cholesky decomposition to solve the resulting linear equation system. Results show that this approximation works well for various models. The details of the formulation is given in Appendix.

### 3.3. Error function for plane

We also use the planar surface as a basic fitting primitive since it is the most commonly used surface type in geometric modeling and processing applications. Both $L^2$ and $L^{2,1}$ distance can be exactly computed for planer surfaces. Given a planar surface represented by $f(\mathbf{x}) = \mathbf{n}^T \cdot \mathbf{x} + d = 0$, $|\mathbf{n}| = 1$. The error function for a plane is then defined by

$$E_d(t, f) = \frac{|t|}{6}(d_1^2 + d_2^2 + d_3^2 + d_1 d_2 + d_2 d_3 + d_3 d_1)$$

$$E_n(t, f) = |t| \cdot |\mathbf{n} - \mathbf{n}_t|^2,$$

where $d_1, d_2, d_3$ are the orthogonal distances from three vertices of $t$ to the plane and $|t|$ is the area of $t$.

## 4. Variational shape segmentation

In this section, we shall describe the implementation details of the presented algorithm. Our segmentation algorithm consists of four main steps: (1) preprocessing; (2) initialization; (3) optimization and (4) postprocessing. The flowchart of our algorithm is shown in Fig. 2.

### 4.1. Preprocessing

In this step, the input mesh $\mathcal{M}$ is first uniformly scaled into the unit cube $[0, 1]^3$. Then we pre-compute the matrix entries for each triangle (see Appendix) which will be used in quadric surface fitting.

### 4.2. Initialization

An initial partition is required to start the global optimization. To initialize, we randomly select $n$ seed faces $\{s_i\}_{i=1}^n$. Then each seed face $s_i$ defines an initial planar proxy $\mathcal{P}_i$ which is the plane passing through the seed face. Then the global optimization starts. Users could also set $n = 1$ at the beginning and then progressively insert new components.

### 4.3. Optimization

The minimization of Eq. (1) is achieved by the Lloyd iteration [6]. The Lloyd iteration terminates when the convergence of Eq. (1) is observed or a maximal number of iteration is reached (30 in all experiments).

#### 4.3.1. Re-grouping

Each time when we have a set of best fitting proxies $\{\mathcal{P}_i\}_{i=1}^n$ and their corresponding seed triangles $\{s_i\}_{i=1}^n$, we want to assign all the triangles to their "nearest" component to drive down the total error. We use the distortion-minimizing flooding algorithm [6] in this step.

The aim of distortion-minimizing flooding is to partition the input mesh into a set of non-overlapping, connected regions $\{\mathcal{R}_i\}_{i=1}^n$. Given a set of seed triangles $\{s_i\}_{i=1}^n$ and their corresponding proxies $\{\mathcal{P}_i\}_{i=1}^n$, we first compute the distance between all the neighboring triangles of each seed triangle $s_i$ and its corresponding proxy $\mathcal{P}_i$. All the tested pairs $(t_j, \mathcal{P}_i)$ are inserted into a global priority queue with a priority equal to $E(t_j, f_i)$ (Eq. (3)). In each step the triangle–proxy pair $(\tilde{t}, \tilde{\mathcal{P}})$ with the smallest distance is popped out from the queue. If $\tilde{t}$ is already assigned to a region we continue the procedure without doing anything; otherwise $\tilde{t}$ is assigned to the region against which it is tested. Then we test all the unlabeled neighboring triangles of $\tilde{t}$ with the current proxy $\tilde{\mathcal{P}}$ and push these new triangle–proxy pairs into the queue. This process is repeated until the queue is empty. Finally we shall get a new partition of the input mesh. The reader is referred to [6] for more details of this algorithm.

#### 4.3.2. Surface fitting

Each time when we have a new partition $\mathcal{R} = \{\mathcal{R}_i\}_{i=1}^n$, we fit a new quadric surface for each region $\mathcal{R}_i$ to minimize the total error again. We first fit a plane to each region $\mathcal{R}_i$, if the fitting error is smaller than a pre-specified tolerance ($1 \times e^{-6}$ in all our experiments), we accept the fitting result and set the surface type of $\mathcal{R}_i$ to plane. Otherwise we fit a quadric surface for this region again. The result with smaller error is accepted. Once the fitting surface for each region is updated, we update the seed face for each region by selecting the face with the smallest error to the fitting surface.

If the termination condition of the Lloyd iteration is not met, then we return to the regrouping step. Otherwise we check whether a new component should be added or the whole optimization algorithm should be terminated.

#### 4.3.3. Region merging and insertion

We also provide operators for region merging and insertion, as done in [6]. When the Lloyd iteration terminates, we detect whether there is any redundant region by testing each pair of adjacent components. A new quadric surface is fitted to each pair of adjacent components $(\mathcal{R}_i, \mathcal{R}_j)$, for which the fitting error is denoted as $E_{i,j}$ and the fitting surface is $f_{i,j}$. A pair of components is tagged as valid if $|E_{i,j} - (E_i + E_j)| < \varepsilon$. If the fitting surface is a *pair of planes* or a *hyperboloid of two sheets* but the projected data points are contained in both sheets, then $f_{i,j}$ is considered as *invalid*, because it is not an appropriate representation [14]. If there are more than one pair of components that are valid, the pair with the smallest increasing error is chosen to be merged, i.e. $\min_{\{\mathcal{R}_i, \mathcal{R}_j\}}(|E_{i,j} - (E_i + E_j)|)$.

If the target number of the regions is fixed, an insertion operation follows after each merging operation, which is called *region teleportation* [6]. Otherwise new region can be inserted directly. We check the validity of the fitting surface of each

**Fig. 3.** Illustration of the segmentation process. The segmentation is initialized with 1 proxy (leftmost), new proxies are progressively inserted. From left to right, 1–6 proxies and final result with 13 proxies (rightmost).

component. A new component $\mathcal{R}_{new}$ will be inserted in the region which is an invalid quadric surface. If all the fitting quadrics are valid, but the total error $E(\mathcal{R})$ is still larger than a pre-specified threshold, a new component (or region) $\mathcal{R}_{new}$ is inserted following the farthest-point criterion, i.e., we find the region $\mathcal{R}_i$ with maximal fitting error $\max_i(\frac{E_i}{|\mathcal{R}_i|})$, where $|\mathcal{R}_i|$ is the total area of component $\mathcal{R}_i$. Then we find the face which has the largest error belonging to $\mathcal{R}_i$ and set the face as the new seed face $s_{new}$. We also provide user interaction tools to insert a new component by indicating a region. The new component $\mathcal{R}_{new}$ is then set to be the plane containing the seed face $s_{new}$. The Lloyd iteration is continued after each merging or insertion step. Fig. 3 illustrates the progressive proxy insertion in the optimization process.

### 4.4. Post-processing

Global optimization terminates when the input mesh is well approximated by a set of quadric surfaces. Although our proposed objective function (Eq. (1)) works well for well structured CAD models, it still results in non-smooth segmentation boundaries, especially for free-form shapes. Hence we propose several post-processing operations to further improve the segmentation quality in this step, including boundary smoothing, simple quadric type identification and proxy projection.

#### 4.4.1. Boundary smoothing

After the global optimization stage, the surface mesh $\mathcal{M}$ has been partitioned into non-overlapping regions $\mathcal{R}_i$, each being fitted by a quadric proxy $\mathcal{P}_i$. Triangle faces next to the segmentation boundary always have nearly equal errors to neighboring proxies, often leading to zigzag boundary curves. The graph cut method has already been used in [27,48,12] to segment mesh in the fuzzy region and boundary regularization, but only dihedral angle and edge length are used in their approach, so it works well mainly in regions with salient features or curvature discontinuity. We propose a new graph cut based strategy which is particularly effective for smoothing boundary curves in a smooth region of the mesh.

Consider the dual graph of the original mesh, each triangle face is corresponding to a dual vertex. Given two neighboring regions $\mathcal{R}_0$ and $\mathcal{R}_1$, the faces belonging to the neighbor of their common boundary are marked as belonging to the fuzzy region (Fig. 4 (left) illustrates the fuzzy region. The neighborhood size can be set by the user). Let $\mathcal{V}_f$ denote the set of the dual vertices of the fuzzy region. Suppose that the faces in the fuzzy region are removed from $\mathcal{R}_0$ and $\mathcal{R}_1$. Then the dual vertices of faces in the regions $\mathcal{R}_0$ and $\mathcal{R}_1$ that are adjacent to $\mathcal{V}_f$ are denoted as $\mathcal{V}_0$ and $\mathcal{V}_1$, respectively.

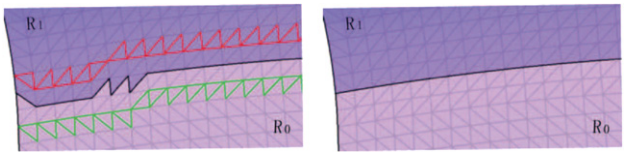


**Fig. 4.** Boundary smoothing. Left: un-smoothed boundary and right: smoothed boundary.

The goal of boundary smoothing is to label the vertices in $\mathcal{V}_f$ with 0 or 1 by minimizing a cost function $E(X)$. This is similar to the binary labeling problem for edge detection widely used in image segmentation. The solution $X$ is a binary vector $X = (x_0, x_1, \ldots)$, $x_i \in \{0, 1\}$. If $v_i \in V_f$ is labeled with 0, i.e., set $x_i = 0$, then its corresponding face is assigned to the region $\mathcal{R}_0$; otherwise, the face is assigned to the region $\mathcal{R}_1$.

Let $G = \{\mathcal{V}, \mathcal{E}\}$ be an undirected sub-graph of the dual graph of the mesh $\mathcal{M}$, where $\mathcal{V} = \mathcal{V}_f \cup \mathcal{V}_0 \cup \mathcal{V}_1$ is the set of nodes. Here $\mathcal{E}$ is the set of undirected edges, with each dual edge $e = (v_i, v_j)$, $(v_i, v_j \in \mathcal{V}, i \neq j)$ corresponding to an edge shared by two adjacent faces in $\mathcal{V}$. In Fig. 4 the background is composed of two regions $\mathcal{R}_0$ and $\mathcal{R}_1$. The set $\mathcal{V}_0$ consists of the green triangles in $\mathcal{R}_0$, the set $\mathcal{V}_1$ consists of the red triangles in $\mathcal{R}_1$, and the set $\mathcal{V}_f$ consists of those triangles between $\mathcal{V}_0$ and $\mathcal{V}_1$. Here $\mathcal{V}_0$ and $\mathcal{V}_1$ are hard constraints to $\mathcal{R}_0$ and $\mathcal{R}_1$ in the sense that the triangles in both sets will keep their labels; only the triangles in $\mathcal{V}_f$ may be re-labeled.

The energy function $E(X)$ is defined in a similar way to [49] used for image segmentation:

$$\begin{aligned} E(X) &= E_1(X) + \lambda E_2(X) \\ &= \sum_{v_i \in \mathcal{V}} \hat{E}_1(x_i) + \lambda \sum_{(v_i, v_j) \in \mathcal{E}} \hat{E}_2(x_i, x_j). \end{aligned}$$

In order to keep the triangle faces in the fuzzy region from deviating too much from their quadric proxies and improve boundary smoothness, we consider both the distance from the boundary faces to their proxies and the edge length along the boundary. The region energy term $E_1$ is determined by how the nodes $v_i$ in $\mathcal{V}_f$ are labeled. Let $d_i^0 = d(v_i, P_0)$ and $d_i^1 = d(v_i, P_1)$ be the distance of $v_i$ to proxies $\mathcal{P}_0$ and $\mathcal{P}_1$. Then we define

$$\hat{E}_1(x_i = 0) = \begin{cases} 0, & v_i \in \mathcal{V}_0 \\ \infty, & v_i \in \mathcal{V}_1 \\ \dfrac{d_i^0}{d_i^0 + d_i^1}, & v_i \in \mathcal{V}_f, \end{cases}$$

$$\hat{E}_1(x_i = 1) = \begin{cases} \infty, & v_i \in \mathcal{V}_0 \\ 0, & v_i \in \mathcal{V}_1 \\ \dfrac{d_i^1}{d_i^0 + d_i^1}, & v_i \in \mathcal{V}_f. \end{cases}$$

The term $\hat{E}_2$ is the cost of a dual edge connecting two adjacent face nodes $\{v_i, v_j\}$, and is defined by

$$\hat{E}_2(x_i, x_j) = \frac{\text{length}(i,j)}{\text{length}(i,j) + \text{ave_length}} |x_i - x_j|,$$

where length$(i, j)$ is the length of the common edge shared by $v_i$ and $v_j$, and ave_length is the average edge length of the mesh $\mathcal{M}$. Clearly, $E_2(X)$ becomes larger when the edge length of the cut boundary resulting from re-labeling is longer. The cost function $E(X)$ is minimized using the max-flow/min-cut algorithm described in [50]. Fig. 4 (right) shows the result of boundary smoothing ($\lambda = 1$ by default).

#### 4.4.2. Quadric surface classification

To simplify the final representation, we would like to identify some commonly used types of special quadrics, such as spheres and circular cylinders, which have occurred as approximating proxies. Given the coefficients of proxy $\mathcal{P}_i$, we detect whether the quadric is nearly a cylinder or a sphere by analyzing the eigenvalues of the corresponding quadratic form [51]. After type identification, the region is fitted by a quadric of the special type that has been identified. Only circular cylinders and spheres are considered as special types in our current implementation.

#### 4.4.3. Proxy projection

As the final step of post-processing, the vertices of each region $\mathcal{R}_i$ of the partitioned mesh $\mathcal{M}$ are projected onto the corresponding proxy $\mathcal{P}_i$ of $\mathcal{R}_i$. The computation of foot points on a plane, sphere or cylinder is straightforward. If the quadric surface belongs to some other types, we compute the exact foot point by solving a 6th-degree univariate equation [51]. For an interior vertex of a region $\mathcal{R}_i$, its projected position is the foot point on the proxy of $\mathcal{R}_i$; if a mesh vertex is shared by two or more regions, the final position is the average of its foot points on all the proxies the vertex belongs to.

## 5. Enhancements

We propose two accelerating techniques in order to efficiently segment models with sharp features or with large size. The improvements include feature-based and simplification-based segmentation, which are optional in our system.

### 5.1. Feature-based segmentation

The input mesh is first pre-partitioned by feature skeletons. In our approach, we simply use the dihedral angle to detect feature edges, but any other feature detection technique can be used instead. All the edges with a dihedral angle exceeding a threshold $\theta_s$ (30° in our implementation) are marked as features. Starting from a randomly selected triangle, we perform greedy region growing to gather neighboring triangles that do not cross sharp edges. This process is repeated until all the triangles are assigned to a group. In the initialization step, we set the initial number of the components to the number of groups formed by feature loops; see Fig. 5(a) & (b) for example of feature edges and pre-partition, respectively. In the partition step, the flooding algorithm is restricted not to cross the feature edges. This simple strategy helps to improve the efficiency of the segmentation for CAD meshes.

### 5.2. Simplification-based segmentation

The mesh simplification technique is an efficient tool to process data with huge size [27,28,52]. In this section, we propose a simplification-based approach to improve the efficiency of our variational segmentation algorithm. The input mesh $\mathcal{M}$ is first simplified to a low resolution version $\mathcal{M}'$ in the preprocessing step. We first segment the simplified mesh $\mathcal{M}'$. Then the segmentation result of $\mathcal{M}'$ is mapped to the original mesh which serves as an initial segmentation. The final result is obtained by optimizing the segmentation again on the original mesh. The process of simplification-based segmentation is illustrated in Fig. 6.

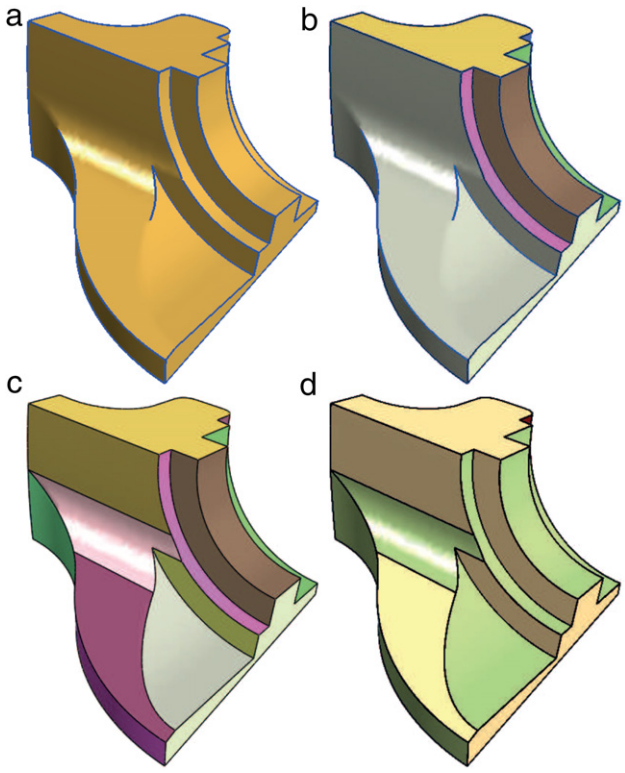


**Fig. 5.** Feature-based segmentation of fandisk model. (a) Input model with feature curves (blue lines); (b) feature-based initialization, 12 initial components; (c) segmentation result, 22 components, taking only half time as compared without using feature information; (d) fitting result, refer to Fig. 1(d) for the meaning of different color. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

## 6. Experimental results

We present the experimental results of our algorithm in this section. The input meshes are assumed to be 2-manifold with arbitrary topology, closed or with open boundaries. All examples are tested on a PC with Intel Xeon 2.66 GHz CPU and 2.00 GB RAM.

Our algorithm works well for tessellated CAD models, which exhibit well defined feature structures. Figs. 1, 3, 5 and 7 demonstrate several such results of feature-based segmentation. The crank model shown in Fig. 1 contains more than 100 K triangles. It takes more than one hour to segment this model without using features, while it takes only minutes for feature-based segmentation. The base of the lamp model in Fig. 7 is fitted by two planes and one cylinder if the feature is taken into account (Fig. 7(c)), otherwise too many proxies will be added since the global cylinder structure is difficult to be detected from low-resolution sampling. Table 1 shows the timing of feature-based segmentation.

Figs. 8 and 14 show two examples of simplification-based segmentation for scanned CAD models. The timing of the simplification-based segmentation is given in Table 2. In all the examples discussed above, we set the normal weight $\omega = 0.5$

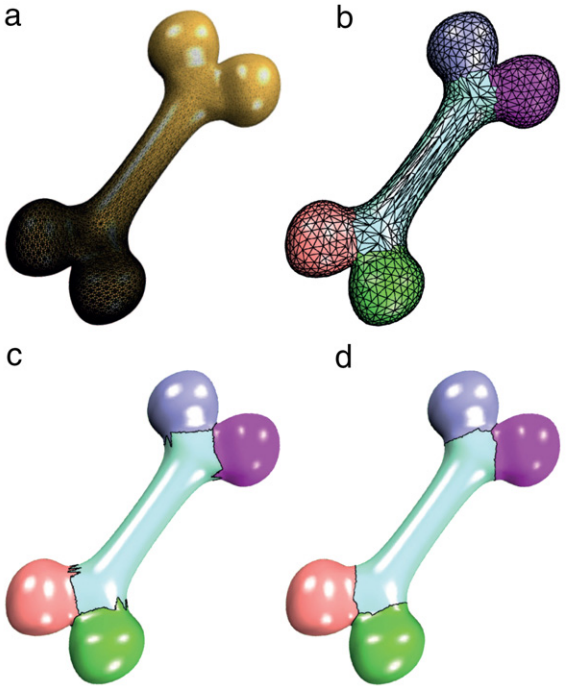


**Fig. 6.** Simplification-based segmentation of bone model. (a) Input model with 30 K faces; (b) segmentation result of simplified mesh (1 K faces) with 5 patches; (c) mapping segmentation result from simplified mesh to original mesh; (d) final segmentation result.

(see Eq. (3)). Notice that we did not apply any boundary smoothing operation.

The proposed variational framework scales well for free-form models. Fig. 6 shows an example on a bone model. However, due to the quadric nature of our fitting algorithm, the segmentation boundaries for organic models are not as smooth as the those of CAD models. Hence we further smooth the zigzag boundaries by a graph-cut based approach (see Section 4.4.1). We show only one example here (see Fig. 9) due to the space limitation, more results can be found in [14].

Comparison. First, we compare our quadric surface fitting algorithm with the previous approach [46]. In their approach, the algebraic distance is used to approximate the exact $L^2$ distance between a triangle and a quadric surface. They use $(\nabla f(\mathbf{x}) - \mathbf{n}_t)^2$ to approximate the normal deviation, where the gradient vector is not normalized. The fitting result of their algorithm forces the gradient to converge to the normalized unit vector. In our approach, we use a combination of the first order approximation of the $L^2$ distance [41] and an approximate normalized gradient to measure the normal deviation. We use several simple examples for comparing our fitting results with that of [46]. As shown in Fig. 11, the input meshes are sampled from several quadric surfaces, i.e., a saddle, an ellipsoid and an elliptic cylinder. It is easy to see that our fitting method obtains better fitting result than [46].

The comparisons with previous variational based approaches [6,10] are given in Fig. 10. We show that our new method obtains more faithful segmentation result than using only simple types of proxy, such as plane, sphere and cylinder. Our algorithm also results in smaller approximation error than previous approach [14]. We use the Metro tool [53] to measure the symmetric Hausdorff distance between the fitting surfaces and input meshes.

We also compare our approach with region merging based method [22]. The region merging based method is fast but the segmentation relies on the local information of the mesh. The segmentation result would be inappropriate in blending regions

**Table 1**
Timing Statistics (in seconds) of feature-based segmentation. $|\mathcal{M}|$ is the number of triangles of input mesh; $|R_i|$ is the number of patches; $T_{nf}$ and $T_f$ are the timings of segmentation with/without using features during the segmentation process, respectively.

| Model | $\lvert\mathcal{M}\rvert$ (K) | $\lvert R_i\rvert$ | $T_{nf}$ | $T_f$ |
|---|---|---|---|---|
| Fandisk | 13 | 22 | 12 | 3 |
| Tesa | 22 | 12 | 19 | 4.7 |
| Pawn | 24 | 12 | 17 | 7.8 |
| Lamp | 38.4 | 54 | 25 | 5 |

**Table 2**
Timing statistics of simplification-based segmentation. $|\mathcal{M}|$ and $|\mathcal{M}'|$ are the number of triangles of input mesh $\mathcal{M}$ and the simplified mesh $\mathcal{M}'$; $|R_i|$ is the number of patches; $T_{nmr}$ and $T_{mr}$ are the timings of segmentation with/without using simplification-based segmentation. The column *Feature* indicates whether the feature is used for segmentation.

| Model | $\lvert\mathcal{M}\rvert$ (K) | $\lvert\mathcal{M}'\rvert$ (K) | Feature | $\lvert\mathcal{R}_i\rvert$ | $T_{nmr}$ | $T_{mr}$ |
|---|---|---|---|---|---|---|
| Fandisk | 13 | 1 | ✓ | 22 | 3 | 1.8 |
| Tesa | 22 | 2 | ✓ | 12 | 4.7 | 2.6 |
| Cover | 13.5 | 2 | × | 3 | 16 | 7 |
| Part2 | 20 | 2 | × | 9 | 19 | 4.9 |
| Part3 | 40 | 2 | × | 11 | 27 | 8.2 |
| Bone | 30 | 3 | × | 5 | 17 | 6.1 |

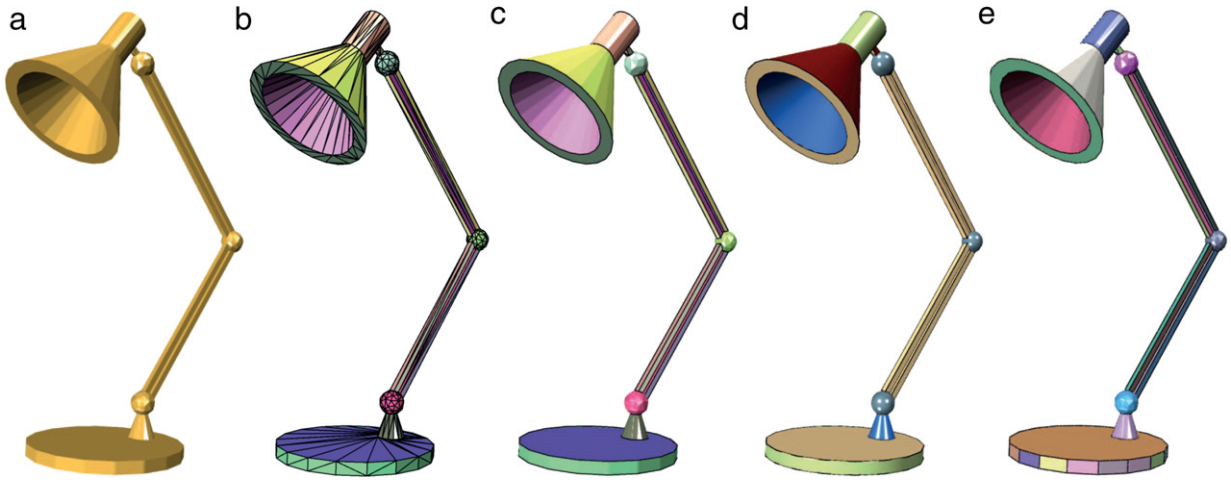


**Fig. 7.** Lamp model: (a) input mesh with 38.4 K faces; (b) segmentation result of simplified model (600 faces) with 45 patches; (c) segmentation result of original model; (d) projecting mesh vertices onto the fitted quadric surfaces; (e) segmentation result without using feature.

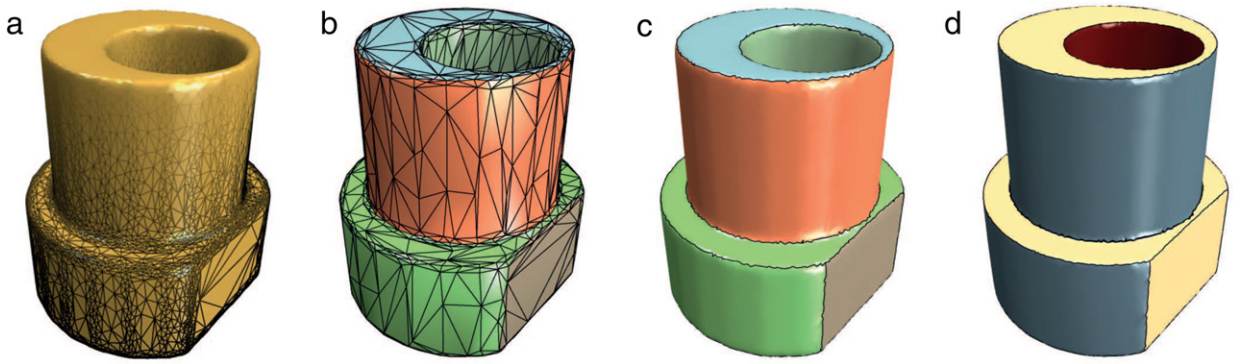


**Fig. 8.** Part2 Model: (a) input mesh with 20 K faces, (b) segmentation result of simplified mesh (2 K faces); (c) segmentation result of original mesh; (d) projecting mesh vertices onto the fitted quadric surfaces.

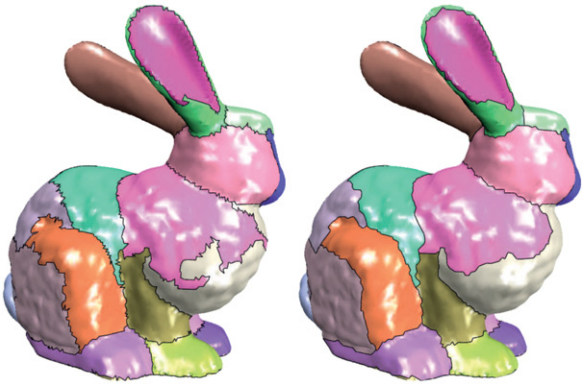

**Fig. 9.** Boundary smoothing of the bunny model: before (left) and after (right) smoothing.

and the regions of complex surfaces except plane, sphere and cylinder. Fig. 12 shows several examples compared with [22]. Fig. 13 shows another comparison with the region merging method [20]. Our algorithm results in more faithful segmentation quality due to its optimization nature.

Limitations. One of the limitations of this work is that the presented quadric surface fitting algorithm is noise sensitive. An example is shown in Fig. 14. In this case, the normal component in object function (Eq. (1)) becomes unreliable, which results in unsatisfied segmentation boundary.

Another limitation is that we are not able to identify other surface types of engineering objects, such as tori and blend/fillet. On the other hand, for the general organic objects, the quadric surfaces may be too flexible. Fig. 15 shows such an example.

## 7. Conclusions and future work

We present an efficient variational framework for mesh segmentation. Each segmented patch is fitted by a quadric surface. Instead of considering only distance or normal deviation, we introduce a new error metric which is a combination of both $L^2$ and $L^{2,1}$ metrics. The new metric results in better segmentation quality without any postprocessing, especially for CAD models. Moreover, the effectiveness of the presented algorithm is demonstrated by various examples and comparison with previous works.

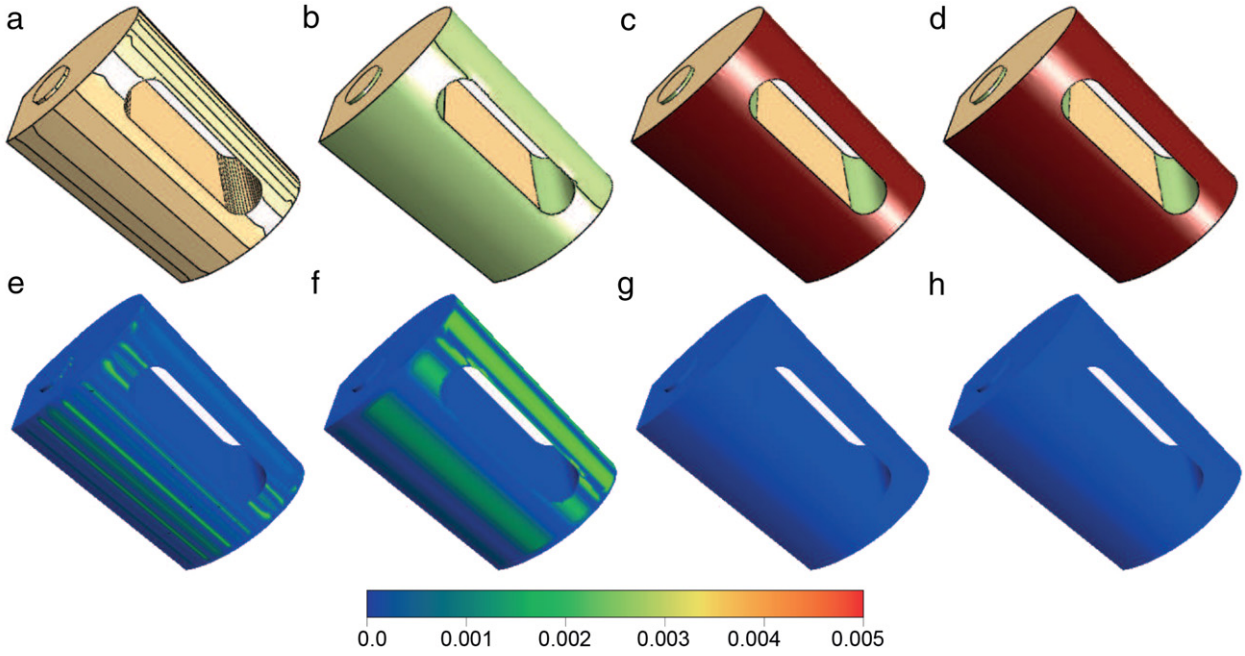


**Fig. 10.** Comparison of approximation error. Top row: segmentation of Tesa model: (a) 80 planar proxies [6]; (b) 24 hybrid proxies [10]; (c) 22 quadric proxies [14] and (d) 22 quadric components by our method. Bottom row: color coding of the RMS Hausdorff errors of corresponding segmentation: (e) $2.1 \times 10^{-2}$ [6]; (f) $2.8 \times 10^{-2}$ [10]; (g) $5.9 \times 10^{-3}$ [14] and (h) ours $3.3 \times 10^{-3}$, respectively.

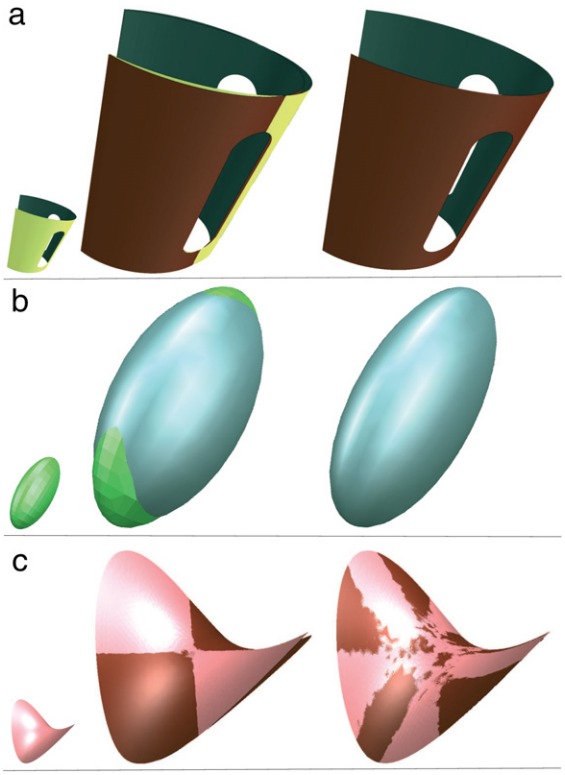


**Fig. 11.** Comparison of quadric surface fitting with [46]. Left column: input mesh surfaces (a) an elliptic cylinder, (b) an ellipsoid and (c) a saddle surface; middle column: results of [46] and right column: results of our method. The fitted surfaces are overlaid with the input meshes.

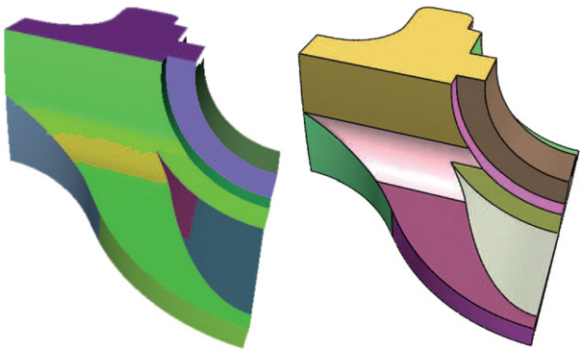

**Fig. 12.** Comparison with hierarchical clustering approach [22]. Left: result of [22] and right: our result.

A number of applications can benefit from the output of our mesh segmentation framework, such as converting mesh surfaces to *B*-Reps in CAD/CAM modeling, efficient collision detection for composite quadric objects. However, we only use the quadric surface as the basic type of proxy in our framework, which is not flexible to identify tori or blends in complex CAD models. In the future, we plan to enrich the types of proxies to handle more complicated inputs. On the other hand, the presented algorithm is not able to capture the global structure of the object, such as symmetries. One interesting problem is to detect global symmetries in the segmentation framework. Some efforts have been made for detecting simple types of symmetric proxies [54,55]. Furthermore, the extension of our algorithm for handling the point cloud data is also of interest.

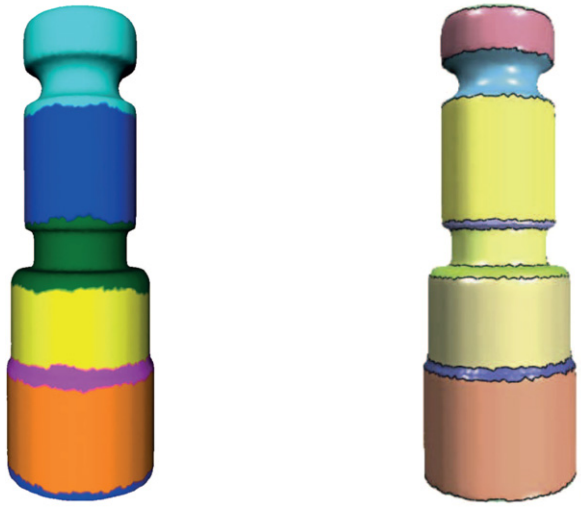

**Fig. 13.** Comparison with [20] (Part3 model). Left: result of [20] and right: our result.

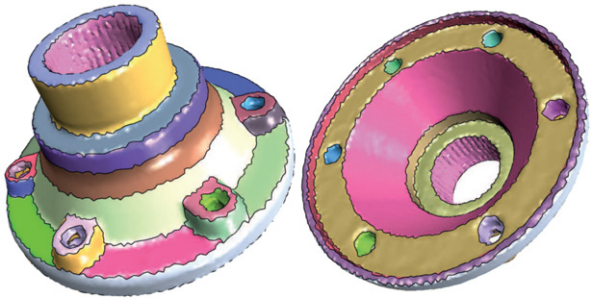

**Fig. 14.** Segmentation of a noisy Carter model (100 K faces), 39 patches.

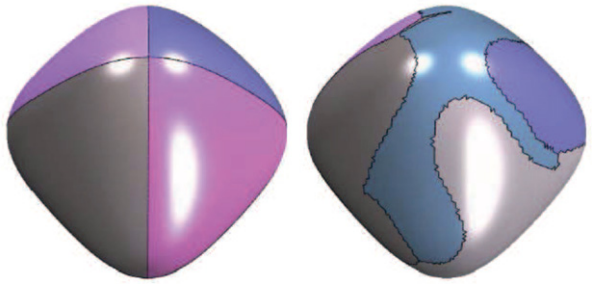

**Fig. 15.** The rounded octahedron model is segmented well by using only the planar proxies (left). The result of using quadric proxies is unsatisfactory (right).

## Acknowledgments

We would like to thank the reviewers for their detailed comments and suggestions which greatly improved the manuscript. We also thank Xiaohong Jia and Yong-Liang Yang for proofreading. This work is partially supported by the Research Grant Council of Hong Kong (718209 and 718010), NSFC (11171322), and ANR/NSFC (60625202, 60911130368) Program.

## Appendix. Entries of quadric surface fitting

Given a quadric surface $f(\mathbf{x}) = \mathbf{C}^T \cdot \mathbf{F} = 0$, where

$\mathbf{x} = [x, y, z]^T$,

$\mathbf{C} = [c_0, c_1, \ldots, c_9]^T$,

$\mathbf{F} = [1, x, y, z, x^2, xy, xz, y^2, yz, z^2]^T$.

The gradient $\nabla f(\mathbf{x}) = [\mathbf{C}^T \cdot \mathbf{f_x}, \mathbf{C}^T \cdot \mathbf{f_z}, \mathbf{C}^T \cdot \mathbf{f_z}]^T$, where

$\mathbf{F}_x = [0, 1, 0, 0, 2x, y, z, 0, 0, 0]^T$,

$\mathbf{F}_y = [0, 0, 1, 0, 0, x, 0, 2y, z, 0]^T$,

$\mathbf{F}_z = [0, 0, 0, 1, 0, 0, x, 0, y, 2z]^T$.

The error function of each region $\mathcal{R}_i$ is

$$E(\mathcal{R}_i, \mathcal{P}_i) = \sum_{t_j \in \mathcal{R}_i} \int_{\mathbf{x} \in t_j} \frac{f(\mathbf{x})^2}{|\nabla f_i^0(\mathbf{c}_{t_j})|^2} \cdot d\sigma + \omega \int_{\mathbf{x} \in t_j} \left( \frac{\nabla f(\mathbf{x})}{|\nabla f_i^0(\mathbf{c}_{t_j})|} - \mathbf{n}_t \right)^2 \cdot d\sigma .$$

In the above formula, the $L^2$ error term can be written as $\tilde{E_d}(\mathcal{R}_i, \mathcal{P}_i) = \mathbf{C}^T \mathbf{A}_d \mathbf{C}$, where

$$\mathbf{A}_d = \sum_{t_j \in \mathcal{R}_i} \mathbf{A}_d^j,$$

$$\mathbf{A}_d^j = \frac{1}{|\nabla f_i^0(\mathbf{c}_{t_j})|^2} \int_{\mathbf{x} \in t_j} (\mathbf{F}\mathbf{F}^T) \cdot d\delta.$$

Similarly, the normal term can be written by $\tilde{E_n}(\mathcal{R}_i, \mathcal{P}_i) = \mathbf{C}^T \mathbf{A}_n \mathbf{C} - 2\mathbf{C}^T \mathbf{B}_n + c_n$, where

$$\mathbf{A}_n = \sum_{t_j \in \mathcal{R}_i} \mathbf{A}_n^j,$$

$$\mathbf{A}_n^j = \frac{1}{|\nabla f_i^0(\mathbf{c}_{t_j})|^2} \int_{\mathbf{x} \in t_j} (\mathbf{F}_x \mathbf{F}_x^T + \mathbf{F}_y \mathbf{F}_y^T + \mathbf{F}_z \mathbf{F}_z^T) \cdot d\delta,$$

$$\mathbf{B}_n = \sum_{t_j \in \mathcal{R}_i} \mathbf{B}_n^j,$$

$$\mathbf{B}_n^j = \frac{1}{|\nabla f_i^0(\mathbf{c}_{t_j})|} \int_{\mathbf{x} \in t_j} (\mathbf{n}_t^x \mathbf{F}_x + \mathbf{n}_t^y \mathbf{F}_y + \mathbf{n}_t^z \mathbf{F}_z) \cdot d\delta,$$

and $c_n$ is a constant. The matrix entries $\mathbf{A}_d^j$, $\mathbf{A}_n^j$ and $\mathbf{B}_n^j$ of each triangle $t_j$ can be pre-computed and stored for later use.

The minimization of the total energy function is to solve the following linear equation system:

$$(\mathbf{A}_d + \omega \mathbf{A}_n)\mathbf{C} = \omega \mathbf{B}_n,$$

which can be solved efficiently by Cholesky decomposition.

## References


[1] Besl PJ, Jain RC. Segmentation through variable-order surface fitting. IEEE Transactions on Pattern Analysis and Machine Intelligence 1988;10(2): 167–92.

[2] Várady T, Martin RR, Cox J. Reverse engineering of geometric models—an introduction. Computer-Aided Design 1997;29(4):255–68.

[3] Várady T, Facello MA. New trends in digital shape reconstruction. In: IMA conference on the mathematics of surfaces. p. 395–412. 2005.

[4] Sander PV, Snyder J, Gortler S, Hoppe H. Texture mapping progressive meshes. In: ACM SIGGRAPH. p. 409–16. 2001.

[5] Lévy B, Petitjean S, Ray N, Maillot J. Least squares conformal maps for automatic texture atlas generation. In: ACM TOG, SIGGRAPH. 21. 3. p. 362–71. 2002.

[6] Cohen-Steiner D, Alliez P, Desbrun M. Variational shape approximation. In: ACM TOG, SIGGRAPH. 23. 3. p. 905–14. 2004.

[7] Marinov M, Kobbelt L. A robust two-step procedure for quad-dominant remeshing. Computer Graphics Forum (EUROGRAPHICS) 2006;25(3):537–46.

[8] Gotsman C, Gumhold S, Kobbelt L. Simplification and compression of 3D meshes. In: Proceedings of the European summer school on principles of multiresolution in geometric modelling. Springer; 2002. p. 319–61.

[9] Lloyd SP. Least square quantization in PCM. IEEE Transactions on Inform Theory 1982;28:129–37.

[10] Wu J, Kobbelt L. Structure recovery via hybrid variational surface approximation. Computer Graphics Forum (EUROGRAPHICS) 2005;24(3):277–84.

[11] Simari P, Singh K. Extraction and remeshing of ellipsoidal representations from mesh data. Graphics Interface 2005;161–8.

[12] Julius D, Kraevoy V, Sheffer A. *D*-charts: quasi-developable mesh segmentation. Computer Graphics Forum (EUROGRAPHICS) 2005;24(3):981–90.

[13] Wang W. Modelling and processing with quadric surfaces. In: Farin G, Hoschek J, Kim M-S, editors. Handbook of computer aided geometric design. Elsevier; 2002. p. 777–95.

[14] Yan D-M, Liu Y, Wang W. Quadric surface extraction by variational shape approximation. In: 4th international conference on geometric modeling and processing—GMP. p. 73–86. 2006.

[15] Petitjean S. A survey of methods for recovering quadrics in triangle meshes. ACM Computing Surveys 2002;34(2):211–62.

[16] Vieira M, Shimada K. Surface mesh segmentation and smooth surface extraction through region growing. Computer-Aided Geometric Design 2005; 22(8):771–92.

[17] Lavoué G, Dupont F, Baskurt A. A new CAD mesh segmentation method, based on curvature tensor analysis. Computer-Aided Design 2005;37(10):975–87.

[18] Jagannathan A, Miller EL. Three-dimensional surface mesh segmentation using curvedness-based region growing approach. IEEE Transactions on Pattern Analysis and Machine Interlligence 2007;29(12):2195–204.

[19] Garland M, Willmott A, Heckbert P. Hierarchical face clustering on polygonal surfaces. In: ACM Symposium on interactive 3D graphics. p. 49–58. 2001.

[20] Gelfand N, Guibas LJ. Shape segmentation using local slippage analysis. In: Symposium on geometry processing. p. 219–28. 2004.

[21] Marinov M, Kobbelt L. Automatic generation of structure preserving multiresolution models. Computer Graphics Forum (EUROGRAPHICS) 2005; 24(3):479–86.

[22] Attene M, Falcidieno B, Spagnuolo M. Hierarchical mesh segmentation based on fitting primitives. The Visual Computer 2006;22(3):181–93.

[23] Geng C, Suzuki H, Yan D-M, Michikawa T, Sato Y, Hashima M, Ohta E. A thin-plate CAD mesh model splitting approach based on fitting primitives. Theory and Practice of Computer Graphics—TPCG 2010;45–50.

[24] Hoffman D, Richards W. Parts of recognition. Cognition 1984;18:65–96.

[25] Hoffman D, Signh M. Salience of visual parts. Cognition 1997;63:29–78.

[26] Mangan AP, Whitaker RT. Partitioning 3D surface meshes using watershed segmentation. IEEE Transactions on Vis. and Computer Graphics 1999;5(4): 308–21.

[27] Katz S, Tal A. Hierarchical mesh decomposition using fuzzy clustering and cuts. In: ACM TOG, SIGGRAPH. 22. 3. p. 954–61. 2003.

[28] Lai Y-K, Hu S-M, Martin RR. Feature sensitive mesh segmentation. In: Proceedings of the ACM Symposium solid and physical modeling—SPM. p. 17–26. 2006.

[29] Lee Y, Lee S, Shamir A, Cohen-Or D, Seidel H-P. Mesh scissoring with minima rule and part salience. Computer-Aided Geometric Design 2005;22(5):444–65.

[30] Zhang H, Liu R. Mesh segmentation via recursive and visually salient spectral cuts. Proceeding of Vision, Modeling, and Visualization 2005;429–36.

[31] Sander PV, Wood Z, Gortler S, Snyder J, Hoppe H. Multi-chart geometry images. In: Symposium on geometry processing. p. 146–55. 2003.

[32] Yamauchi H, Gumhold S, Zayer R, Seidel HP. Mesh segmentation driven by Gaussian curvature. The Visual Computer 2005;21(8–10):649–58.

[33] Yamauchi H, Lee S, Lee Y, Ohtake Y, Belyaev A, Seidel H-P. Feature sensitive mesh segmentation with mean shift. Shape Modeling International—SMI 2005;236–43.

[34] Carr N, Hoberock J, Crane K, Hart JC. Rectangular multi-chart geometry images. In: Symposium on geometry processing. p. 181–90. 2006.

[35] Shatz I, Tal A, Leifman G. Paper craft models from meshes. The Visual Computer 2006;22(9–11):825–34.

[36] Yan D-M, Wintz J, Mourrain B, Wang W, Boudon F, Godin C. Efficient and robust reconstruction of botanical branching structure from laser scanned points. 11th IEEE International Conference on Computer-Aided Design and Computer Graphics – CAD/Graphics 2009;572–5.

[37] Ahn SJ, Rauh W, Cho HS, Warnecke HJ. Orthogonal distance fitting of implicit curves and surfaces. IEEE Transactions on Pattern Analysis and Machine Interlligence 2002;24(5):620–38.

[38] Atieg A, Watson GA. A class of methods for fitting a curve or surface to data by minimizing the sum of squares of orthogonal distances. Journal of Computational and Applied Mathematics 2003;158(2):277–96.

[39] Bookstein FL. Fitting conic sections to scattered data. Computer Graphics and Image Processing 1979;9:66–71.

[40] Chen Y-H, Liu C-Y. Quadric surface extraction using genetic algorithms. Computer-Aided Design 1999;31(1):101–10.

[41] Taubin G. Estimation of planar curves, surfaces and nonplanar space curves defined by implicit equations with applications to edge and range image segmentation. IEEE Transactions on Pattern Analysis and Machine Interlligence 1991;13(11):1115–38.

[42] Ohtake Y, Belyaev A, Alexa M, Turk G, Seidel H-P. Multi-level partition of unity implicits. In: ACM TOG, SIGGRAPH. 23. 3. p. 463–70. 2003.

[43] Nehab D, Rusinkiewicz S, Davis J, Ramamoorthi R. Efficiently combining positions and normals for precise 3D geometry. In: ACM TOG, SIGGRAPH. 24. 3. p. 536–43. 2005.

[44] Shen C, O'Brien JF, Shewchuk JR. Interpolating and approximating implicit surfaces from polygon soup. In: ACM TOG, SIGGRAPH. 23. 3. p. 896–904. 2004.

[45] Ohtake Y, Belyaev A, Alexa M. Sparse low-degree implicit surfaces with applications to high quality rendering, feature extraction, and smoothing. In: Symposium on geometry proccessing. p. 149–58. 2005.

[46] Kanai T, Ohtake Y, Kase K. Hierarchical error-driven approximation of implicit surfaces from polygonal meshes. In: Symposium on geometry processing. p. 21–30. 2006.

[47] Taubin G. An improved algorithm for algebraic curve and surface fitting. International conference on computer vision—ICCV. p. 658–65. 1993.

[48] Katz S, Leifman G, Tal A. Mesh segmentation using feature point and core extraction. The Visual Computer 2005;21(8–10):649–58.
[49] Li Y, Sun J, Tang CK, Shum HY. Lazy snapping. In: ACM TOG, SIGGRAPH. 23. 3. p. 303–08. 2004.
[50] Boykov Y, Kolmogorov V. An experimental comparison of min-cut/max-flow algorithms for energy minimization in vision. IEEE Transactions on Pattern Analysis and Machine Interlligence 2004;26(9):1124–37.
[51] Fitzgibbon AF, Eggert DW, Fisher RB. High-level CAD model acquisition from range images. Computer-Aided Design 1997;29(4):321–30.
[52] Ji Z, Liu L, Chen Z, Wang G. Easy mesh cutting. Computer Graphics Forum (EUROGRAPHICS) 2006;25(3):283–91.
[53] Cignoni P, Rocchini C, Scopigno R. Metro: measuring error on simplified surfaces. Computer Graphics Forum 1998;17(2):167–74.
[54] Podolak J, Golovinskiy A, Rusinkiewicz S. Symmetry-enhanced remeshing of surfaces. In: Symposium on geometry processing. p. 235–42. 2007.
[55] Li Y, Wu X, Chrysathou Y, Sharf A, Cohen-Or D, Mitra NJ. GlobFit: consistently fitting primitives by discovering global relations. In: ACM TOG, SIGGRAPH. 23. 3. p. 52:1–12. 2011.