Quadric Surface Extraction
by Variational Shape Approximation

Dong-Ming Yan, Yang Liu, and Wenping Wang

The University of Hong Kong, Pokfulam Road, Hong Kong, China
{dmyan, yliu, wenping}@cs.hku.hk

Abstract. Based on Lloyd iteration, we present a variational method for extracting general quadric surfaces from a 3D mesh surface. This work extends the previous variational methods that extract only planes or special types of quadrics, i.e., spheres and circular cylinders. Instead of using the exact L^2 error metric, we use a new approximate L^2 error metric to make our method more efficient for computing with general quadrics. Furthermore, a method based on graph cut is proposed to smooth irregular boundary curves between segmented regions, which greatly improves the final results.

Keywords: variational surface approximation, quadric surface fitting, graph cut, segmentation.

1 Introduction

Polygonal mesh surfaces are an important shape representation of complex 3D models, now readily acquired with 3D digital scanners or derived from CT/MRI volume data. But a high level concise and faithful geometric representation of mesh data is always desirable for geometry processing or rendering in graphics and CAD/CAM.

The Lloyd method for data clustering is employed in [1] to generate piecewise planar approximation of mesh surfaces. Each planar facet is called a *proxy* representing the part of the mesh surface approximated by the facet. This approach is extended by Wu and Kobbelt [2] to include spheres, circular cylinders and rolling-ball surfaces as additional types of proxies to achieve a more compact approximation. The confinement to these special surfaces is largely due to the relative ease of computing the exact Euclidean distance from a point to such surfaces.

There are two contributions in the present paper. Firstly, motivated by wide application and superior approximation power of quadrics, within the same clustering framework, we further extend the surface types of proxies to include general quadric surfaces, or *quadrics* for short, plus planes. We show how Euclidean distance from a triangle to a quadric can be computed in an approximate but efficient manner, while delivering robust segmentation results.

Secondly, we propose a new method for smoothing irregular boundary curves between adjacent segmented regions through energy minimization using a graph cut approach. This step produces more regular boundary curves, resulting in significantly improved segmentation results compared against previous results (e.g., [2]).

1.1   Related Work

There are two areas of research that are closely related to our work: *shape approxima-tion* and *mesh segmentation.* The main purpose of shape approximation is to compute a simple and compact surface representation of a complex geometric shape, based on different surface types or different computational approaches. We will mainly review those methods that employ the clustering approach.

Shape Approximation. Cohen-Steiner et. al [1] propose a shape approximation al-gorithm based on clustering approach to optimally approximate a mesh surface by a specified number of planar faces. This optimization problem is solved as a discrete partition problem using the Lloyd algorithm [3], which is commonly used for solving the k-mean problem in data clustering. There are two iterative steps in this method: *mesh partition* and *fitting a plane face*, called a *proxy*, to each partitioned region. This method proves effective especially for extracting features and planar regions, but tends to produce an overly large number of planar proxies for a good approximation of a free-form surface. Because of its optimization nature, the method is often referred to as a *variational method*.
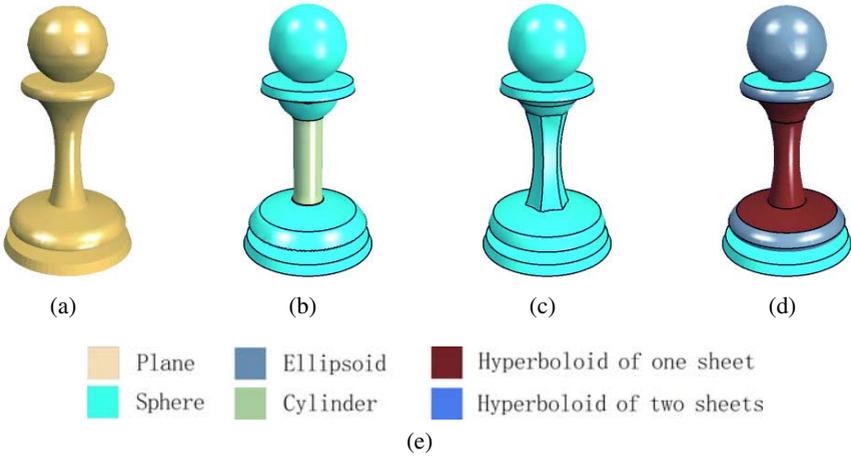

Fig. 1. (a) Original Chess piece (12K triangles); (b) Approximated by 12 hybrid proxies by the method in [2]: 1 plane, 1 cylinder and 10 spheres; (c) Approximated by 18 hybrid proxies by the method in [2]: 1 planes and 17 spheres; (d) Approximated by 12 quadric proxies by our method: 1 plane, 4 spheres, 4 ellipsoids, and 3 hyperboloids of one sheet; (e) Colors for different types of quadric surfaces used in this paper

Wu and Kobbelt [2] extend the work in [1] by introducing spheres, cylinders and rolling ball patches as additional basic proxy types, so that a complex shape can be approximated to the same accuracy by a much fewer number of proxies, leading to a more compact representation. However, these newly added surface types mentioned are still rather restricted, even for CAD models and other man-made objects. For example, the middle part of the Chess piece in Fig. 1 cannot be well approximated either by a circular cylinder or a collection of spherical surface strips.

Simari et al. [4] use ellipsoids as the only type of proxies for approximating mesh surfaces, again using the Lloyd method with the error metric being a combination of Euclidean distance, angular distance and curvature distance. The segmentation boundaries are smoothed by a constrained relaxation of the boundary vertices. They also approximate the volume bounded by a mesh surface using a union of ellipsoids, where whole ellipsoids, rather than ellipsoidal surface patches, are used. Julius et. al [5] segment mesh surfaces into developable surface charts for texture mapping and pattern design. Open segmentation boundaries are straightened by a shortest path algorithm and interior segmentation boundaries are smoothed by a graph cut method similar to that described in [6].

Attene et al. [7] propose a fast algorithm using automatic face clustering to segment a mesh hierarchically. A binary cluster tree is created from bottom to top. At each iteration, every pair of adjacent clusters are fitted by plane, sphere and cylinder, the pair with the minimal fitting error is merged into one cluster. Smoothing of segmentation boundaries is not considered.

Implicit surfaces have long been used for shape approximation and segmentation. Based on *region growing*, Besl et al. [8] segment range image data by fitting implicit surfaces of variable orders. Their algorithm works well on objects with sharp features or curvature discontinuities, but cannot handle free-from shapes. Fitzgibbon et al. [9] improve this work, also using region growing, to fit general quadric surfaces and planes to the range images, and compute surface intersections to extract a B-rep from the segmented image.

Since region growing relies mainly on local consideration, such as mean curvature and Gaussian curvature estimation, the segmentation result can be poor when there is no obvious curvature discontinuity, e.g., when two quadric patches join with near G^1 continuity. In this regard the iterative variational method has a distinct advantage that the local error in partition can be corrected by the fitting process, and the improved partition in turn provides a more reliable basis for better fitting.

Mesh Segmentation. Besides being used as a preparatory step for surface approximation, mesh segmentation is also used to partition a surface model into meaningful parts for various other purposes [10, 11, 12, 13, 6, 14, 15, 7, 16]. A detailed discussion of mesh segmentation methods is out of the scope of this paper. We refer the reader to the survey in [17].

Two recent methods for mesh segmentation are worth mentioning. Katz et al. [6] use fuzzy clustering and graph cut to segment a mesh. The mesh is first clustered by the geodesic distance. A fuzzy region is created between every two adjacent components. Finally the fuzzy region is segmented by a graph cut method to yield a smooth boundary. Lavoué et al. [16] present a mesh segmentation algorithm based on curvature tensor analysis. The mesh is first decomposed into several patches, each patch with nearly constant curvature. Then the segmentation boundary is rectified based on the curvature tensor directions.

Smoothing boundaries between adjacent segmented regions is usually considered as a post-processing step after mesh segmentation [6, 15, 16, 5]. We propose a new graph cut based strategy to smooth the segmentation boundary, which considers both the approximation error and the smoothness of the boundary between neighbor regions, and delivers better results in smooth regions of a surface (e.g., see Fig. 2(d)).

2   Preliminary

In this section we describe the variational shape approximation framework and introduce a new error function for measuring the distance between a mesh surface and a quadric surface.

2.1   Variational Framework

Let $\mathcal{M}$ denote an input mesh surface, and $\mathcal{T}$ denote the set of triangles of $\mathcal{M}$. Suppose that $\mathcal{M}$ is partitioned into n non-overlapping regions, denoted as $\mathcal{R} = \{\mathcal{R}_i\}_{i=1}^n$, each region $\mathcal{R}_i$ containing a set of triangles $\mathcal{T}_i = \{t_k^i\}_{k=1}^{n_i}$ such that $\bigcup_{i=1}^n \mathcal{T}_i = \mathcal{T}$. Each region is approximated by a quadric proxy (including the plane as a special case). A quadric proxy, denoted as $\mathcal{P}_i$, is represented by the coefficients of its associated quadratic form. A seed face, denoted as $\mathcal{S}_i$, is a triangle face in $\mathcal{T}_i$ that has the smallest error to the quadric proxy $\mathcal{P}_i$.

In a variational framework the optimal partition $\mathcal{R} = \{\mathcal{R}_i\}_{i=1}^n$ is found by minimizing the following objective function [1, 2]:

$$E(\mathcal{R}, \mathcal{P}) = \sum_{i=1}^n E'(\mathcal{R}_i, \mathcal{P}_i) = \sum_{i=1}^n \sum_{k=1}^{n_i} d(t_k^i, \mathcal{P}_i), \tag{1}$$

where $d(t_k^i, \mathcal{P}_i)$ measure the error between the triangle t_k^i and the proxy $\mathcal{P}_i$. Therefore, $E'(\mathcal{R}_i, \mathcal{P}_i)$ is the error between the partitioned region $\mathcal{R}_i$ and its approximating proxy $\mathcal{P}_i$. The error terms used in our method will be defined in Section 2.2.

Lloyd's algorithm minimizes the above objective function through iterative partition and fitting. Given a specified number n of proxies, the surface mesh $\mathcal{M}$ is first partitioned into n non-overlapping regions. Then the two alternative steps of quadric surface fitting and region partitioning are performed iteratively to reduce the value of the objective function until convergency or a specified number of iterations is reached. More details about this framework can be found [1].

For initialization, we randomly choose n initial seed faces. Then each seed face determines a planar proxy which is the plane containing the seed face. Then a distortion-minimizing flooding is performed, as described in [1], to give an initial partitioned mesh consisting of n regions $\mathcal{R} = \{\mathcal{R}_i\}_{i=1}^n$.

2.2   Error Metric for Proxies

The objective function in the variational shape approximation framework is defined in terms of error terms. Both $L^{2,1}$ and L^2 metrics have been tested in [1, 2]. For planar proxies, it is possible to derive closed formulas of L^2 and $L^{2,1}$ error terms, and $L^{2,1}$ proves to produce better results. Wu and Kobbelt [2] use an approximate L^2 error term to measure the distance from a triangular face to a *hybrid proxy*, which is a sphere, a circular cylinder or a rolling ball blending surface, expressed in terms of the exact L^2 distances from the three vertices of the triangle to the proxy.

While it is easy to compute the Euclidean distance from a point to a sphere or a circular cylinder, it is not desirable to use the same error term as in [2] when extending proxy

types to general quadric surface, because computing the exact distance from a point to a quadric involves solving for the roots of a degree six univariate polynomial. This is a very time consuming task, because the distance computation needs to be performed for many vertices/triangles of a large mesh in many iterations – according to our test on an implementation based on computation of the exact Euclidean distance, each iteration takes about 20 seconds for a mesh of 10K vertices on a PC with a Xeon(TM) 2.66 GHz CPU. This would render our method too inefficient. We have also tested both algebraic distance $|f|$ and first-order approximation $\frac{|f|}{\|\nabla f\|}$ in our algorithm. But they turned out not to work well due to relatively large approximation errors.

Balancing efficiency and accuracy, we choose to use Taubin's second order approximation of the Euclidean distance $\delta_d(\mathbf{p}, \mathbf{Z}(f))$ [18] from the point $\mathbf{p}$ to the surface $\mathbf{Z}(f)$, which is the zero set of the function f. The function $f(x, y, z)$ is given as:

$$f(x, y, z) = C_0 + C_1 x + C_2 y + C_3 z + C_4 x^2 + C_5 xy + C_6 xz + C_7 y^2 + C_8 yz + C_9 z^2. \quad (2)$$

This approximate distance $\delta_d(\mathbf{p}, \mathbf{Z}(f))$ has the favorable property that it is bounded between 0 and $d(\mathbf{p}, \mathbf{Z}(f))$. For quadric surfaces, $\delta_d(\mathbf{p}, \mathbf{Z}(f))$ is given by the only non-negative root of a quadratic polynomial $g(t) = at^2 + bt + c$, where

$$a = - \left[\|C_1 + <2C_4, C_5, C_6> \cdot p\|^2 + \|C_2 + <C_5, 2C_7, C_8> \cdot p\|^2 + \right.$$
$$\left. \|C_3 + <C_6, C_8, 2C_9> \cdot p\|^2 \right]^{1/2},$$
$$b = - \left[(C_5^2 + C_6^2 + C_8^2)/2 + C_4^2 + C_7^2 + C_9^2 \right]^{1/2},$$
$$c = |f(p)|. \quad \textbf{\textcolor{red}{Errata: a and b should be exchanged here !}}$$

Based on this approximate distance, the approximated L^2 distance for a triangle t_j with respect to a quadric surface $\mathcal{P}_i$ is defined as:

$$d(t_j, \mathcal{P}_i) = \frac{1}{m} \sum_{k=1}^{m} \delta_d(\mathbf{p}_k, \mathbf{Z}_i(f))^2 \cdot A_j, \quad (3)$$

where $\{\mathbf{p}_k\}_{k=1}^m$ are uniformly sampled points on the triangle t_j, and A_j, the area of t_j, serves as a weighting factor to account for triangles of different sizes. In implementation we set $m = 4$, i.e., use the vertices and the barycenter point of t_i, and have obtained satisfactory results.

The approximated L^2 distance between $\mathcal{R}_i$ and $\mathcal{P}_i$ is then defined as:

$$E'(\mathcal{R}_i, \mathcal{P}_i) = \sum_{t_j \in \mathcal{R}_i} d(t_j, \mathcal{P}_i) / \sum_{t_j \in \mathcal{R}_i} A_j. \quad (4)$$

We use $\widetilde{L}^2$ to denote this approximation to L^2 distance. To have a uniform comparison, all mesh surfaces are scaled uniformly to fit in a rectangular box with their longest side being 1.

2.3 Quadric Surface Fitting

Given a region $\mathcal{R}_i$, we need to fit a quadric surface to $\mathcal{R}_i$ in L^2 metric. Two common ways of fitting implicit surfaces are the algebraic distance-based fitting and orthogonal

distance-based fitting [19]; in general, the latter produces better fitting results than the former but is computationally much more costly. Since surface fitting is performed repeatedly in our present application, it is not necessary to take much time to compute the best fitting result in L^2 metric in each single intermediate iteration. Hence, we use Taubin's method [20] based on a first-order approximation of L^2 metric for quadric surface fitting.

Let $f(x, y, z) = 0$ be a quadric surface (see Eqn. 2). The squared distance from a point $\mathbf{p}$ to the implicit surface $Z(f) = \{(x, y, z) | f(x, y, z) = 0, x, y, z \in \mathbb{R}\}$ is approximated as $d(\mathbf{p}, \mathbf{Z}(f))^2 \approx \dfrac{f(\mathbf{p})^2}{\|\nabla f(\mathbf{p})\|^2}$. The original method proposed by Taubin is applied to a set of data points. Given a set of points $\{\mathbf{p}_i, i = 1, \cdots, n\}$, the sum of approximated squared distance is, following [20]

$$\frac{1}{n} \sum_{i=1}^{n} d(\mathbf{p}_i, \mathbf{Z}(f))^2 \approx \frac{\frac{1}{n} \sum_{i=1}^{n} f(\mathbf{p}_i)^2}{\frac{1}{n} \sum_{i=1}^{n} \|\nabla f(\mathbf{p}_i)\|^2} = \frac{\mathbf{s}^T M \mathbf{s}}{\mathbf{s}^T N \mathbf{s}},$$

where M, N are coefficient matrices and $\mathbf{s} = < C_0, C_1, \ldots, C_9 >^T$.

For our application, we need to treat the data points as the continuum of surface points distributed uniformly over the mesh surface. Therefore, we adapt Taubin's method by defining the sum of approximated squared distance as follows:

$$\frac{1}{A} \sum_{k=1}^{n_i} \int_{t_k} d(\mathbf{p}, \mathbf{Z}(f))^2 \mathrm{d}\mathbf{p} \approx \frac{\frac{1}{A} \sum_{k=1}^{n_i} \int_{t_k} f(\mathbf{p})^2 \mathrm{d}\,\mathbf{p}}{\frac{1}{A} \sum_{k=1}^{n_i} \int_{t_k} \|\nabla f(\mathbf{p})\|^2 \mathrm{d}\,\mathbf{p}} = \frac{\mathbf{s}^T M_t \mathbf{s}}{\mathbf{s}^T N_t \mathbf{s}}, \qquad (5)$$

where M_t, N_t are coefficient matrices, and A is the sum of the areas of all the triangles in $\mathcal{R}_i$. Hence, the fitting problem is reduced to computing the eigenvector of $M_t - \lambda N_t$ associated with the minimum eigenvalue [20].

Although, due to efficiency consideration, only approximations to the L^2 error metric (i.e., the true squared Euclidean distance) are used in our fitting and partitioning steps, we find that this treatment works robustly and efficiently in practice with the variational shape approximation framework.

3   Variational Quadric Approximation

There are two stages in our method: *global optimization* and *post-processing*. In first stage, the surface $\mathcal{M}$ is partitioned into regions $\mathcal{R}_i$ iteratively according to the error metric defined in the preceding section. Then the boundary curves between neighboring regions are smoothed using a graph cut method and an approximate surface is created by projecting mesh vertices onto their proxies. Fig. 2 illustrates some intermediate steps of our algorithm.
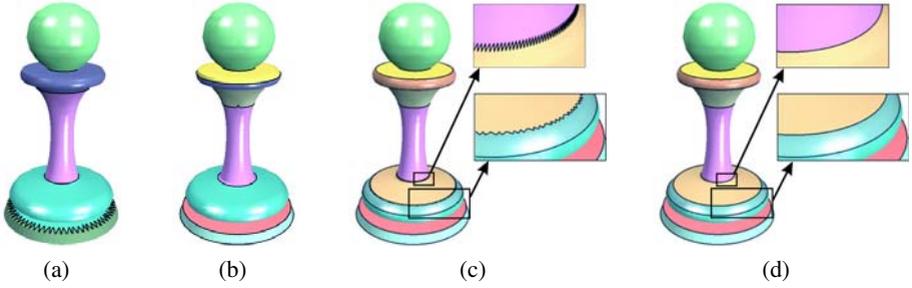
<center>(a)   (b)   (c)   (d)</center>

Fig. 2. Intermediate results of our method. New proxies are inserted progressively and the final projected result is shown in Fig. 1(d). (a) Result when the proxy number is 5; (b) Result when the proxy number is 9; (c) Lloyd iteration finishes when proxy number is 12; (d) The boundaries in (c) are smoothed.

3.1   Global Optimization

The main idea of Lloyd iterations and the initialization have been discussed in Section 2. Typically we choose the number of regions $n = 1$ at the beginning and the new proxies are inserted progressively. The algorithm terminates when an error threshold is met and the Lloyd iteration has converged. Proxies are inserted or merged to achieve the optimal approximation, as described below. Other proxy operations, such as proxy deletion or teleportation, have also been implemented, as in [1].

Proxy Insertion. When the Lloyd iteration has converged, we need to check the validity of each quadric proxy. If the quadric surface is a pair of planes or a *hyperboloid of two sheets* and the projected data points are contained in both sheets, then the proxy is considered as *invalid*, because it is not an appropriate representation. If such a case occurs, new proxies will be inserted in such a region. Fig. 3 gives an example of a region fitted by one degenerate quadric consisting of two intersecting planes, which needs to be split into two separate planar proxies.

   If every proxy is valid but the fitting error is still larger than a pre-specified threshold, we use the farthest-point criterion to add a new proxy $\mathcal{P}_{new}$; that is, we first find the region with maximal fitting error and a face which has the largest $\widetilde{L}^2$ error belonging to this region as the seed face $\mathcal{S}_{new}$. The new proxy $\mathcal{P}_{new}$ is then set to be the plane containing the seed face $\mathcal{S}_{new}$. Then the Lloyd iteration is continued.
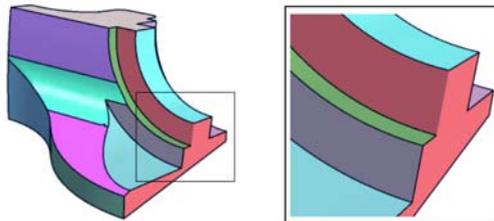

Fig. 3. Left: The red region is fitted by a degenerate quadric consisting of two intersecting planes; Right: Close-up view

Proxy Merging. When the Lloyd iteration converges, we also check whether there are redundant proxies by considering merging each pair of adjacent regions $\mathcal{R}_i$ and $\mathcal{R}_j$. We use a quadric surface $\mathcal{P}_t$ to fit the union of $\mathcal{R}_i$ and $\mathcal{R}_j$. Let E_t be the fitting error. If $|E_t - (E'(\mathcal{R}_i, \mathcal{P}_i) + E'(\mathcal{R}_j, \mathcal{P}_j))| < \epsilon$ (we set $\epsilon = 0.5 * \max_i E'(\mathcal{R}_i, \mathcal{P}_i)$, as in [1]), and $\mathcal{P}_t$ is a valid proxy, then $\mathcal{R}_i$ and $\mathcal{R}_j$ are merged into one region. If there are several pairs can be merged at the same time, the pair with the smallest fitting error is chosen to be merged first.

3.2   Post-processing

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 region boundary curves always have nearly equal $\widetilde{L}^2$ errors, often leading to zigzag boundary curves. The graph cut method has already been used in [6, 15, 5] 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(a) illustrates the fuzzy region. The size of neighbor 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.

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, if $x_i = 1$, 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

(a)   (b)

Fig. 4. Boundary smoothing. (a) Un-smoothed boundary; (b) Smoothed boundary.

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 [21] used for image segmentation:

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

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 $\widetilde{L}^2$ distance of v_i to proxies $\mathcal{P}_0$ and $\mathcal{P}_1$. Then we define

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

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

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

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 [22]. Fig. 4(b) shows the result after running the max-flow algorithm.

For different mesh models, two parameters in the above smoothing algorithm can be set by the user in order to obtain satisfactory results. For the CAD models with clear structure, such as the Fandisk and the Chess piece, the one ring neighborhood of the boundary usually suffices; and we use $\lambda = 1.0$ for this type of models in all the examples presented in this paper. For the free-form objects like the Bunny, Homer, the two- or three-ring neighbor of the boundary works well in our experiments; and the value of λ is selected by the user in the interval $[1, 10]$.

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 [9]. 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.
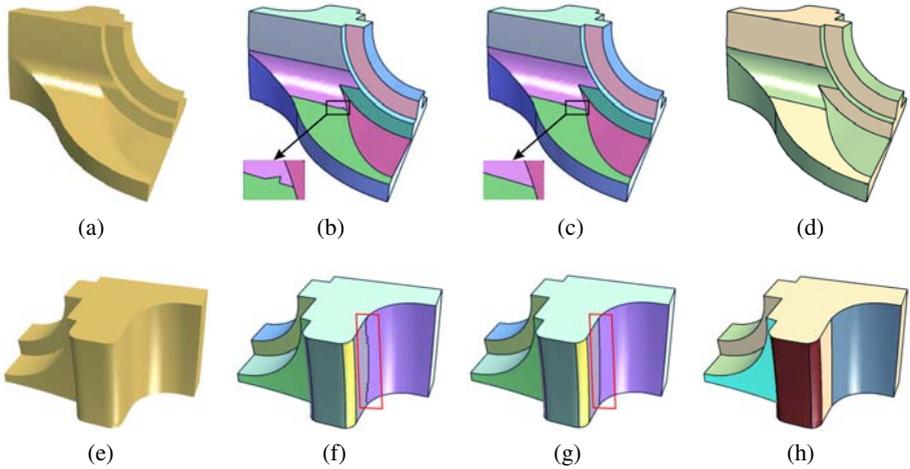
Fig. 5. Fandisk (13K triangle faces): Two views of observation. (a)&(e) The original model; (b)&(f) Partitioned by 22 quadric proxies; (c)&(g) Result after boundary smoothing ($\lambda = 1.0$); (d)&(h) Vertices projected onto proxies.
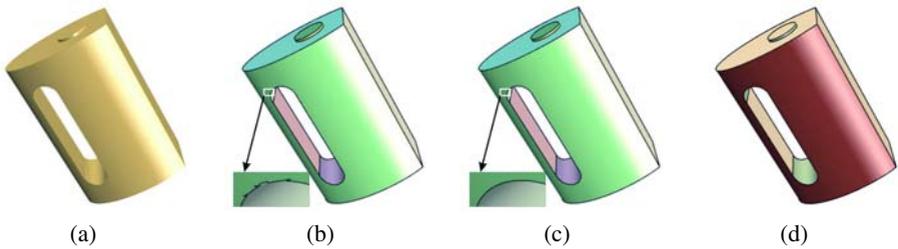

Fig. 6. Tesa [23] (22K triangle faces): (a) The original model; (b) Partitioned by 12 quadric proxies; (c) Result after boundary smoothing ($\lambda = 1.0$); (d) Vertices projected onto proxies
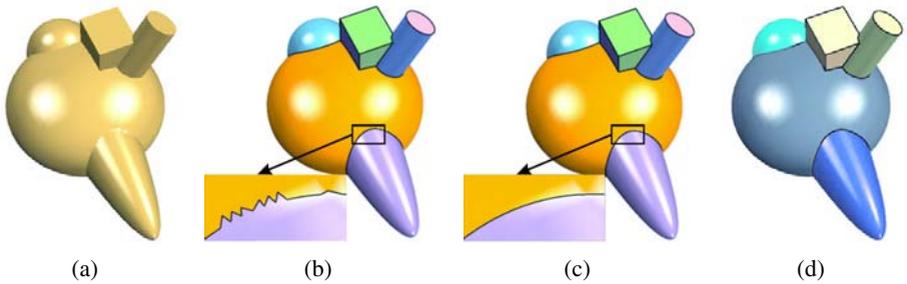

Fig. 7. CSG model (21K triangle faces): (a) The original model; (b) Partitioned by 10 quadric proxies; (c) Result after boundary smoothing ($\lambda = 1.0$); (d) Mesh vertices projected onto proxies

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 degree six univariate equation [9]. 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.

4 Results

In this section we present some test examples to show the effectiveness of our method and to compare it with some previous methods that are also based on variational shape approximation. These examples are shown in Fig. 5, 6, 7 and 9. The meaning of colors
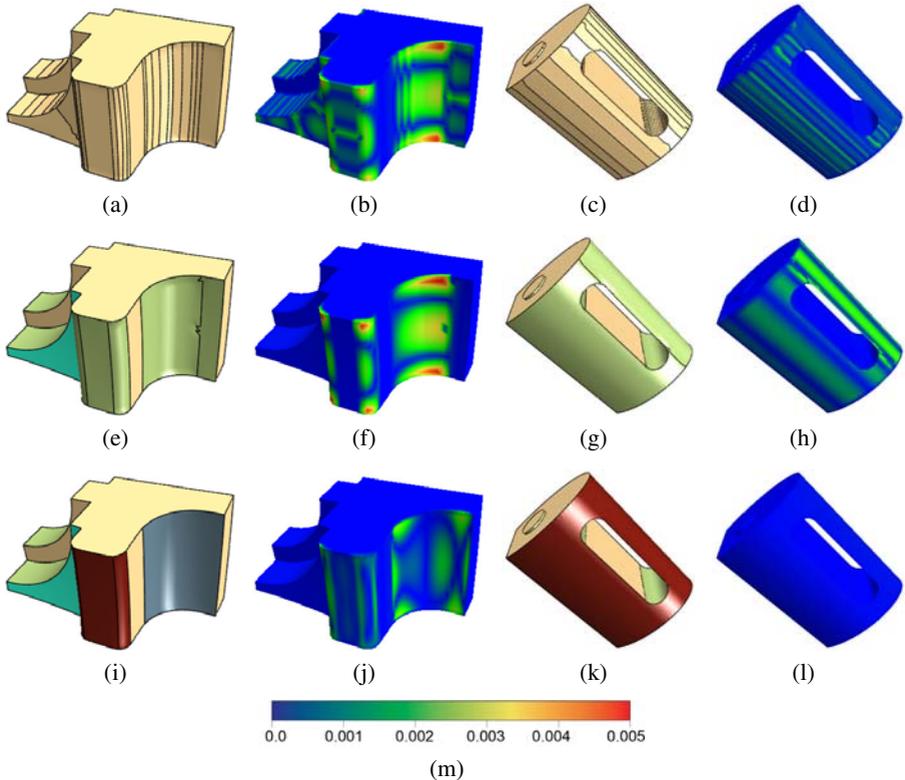

Fig. 8. Comparison with previous methods (from top to bottom). *Column 1*: Findisk model approximated by 80 planar proxies [1], 24 hybrid proxies [2] and 22 quadric proxies by our method; *Column 2*: Color coding of local errors between the approximated model and the original model. The RMS Hausdorff errors are 4.2×10^{-2}, 3.9×10^{-2} and 2.1×10^{-2}, respectively; *Column 3*: Tesa model approximated by 100 planar proxies [1], 14 hybrid proxies [2] and 12 quadric proxies by our method; *Column 4*: Color coding of the local errors. The RMS Hausdorff errors are 3.4×10^{-2}, 4.8×10^{-2} and 4.0×10^{-3}, respectively; (m) Color error bar.
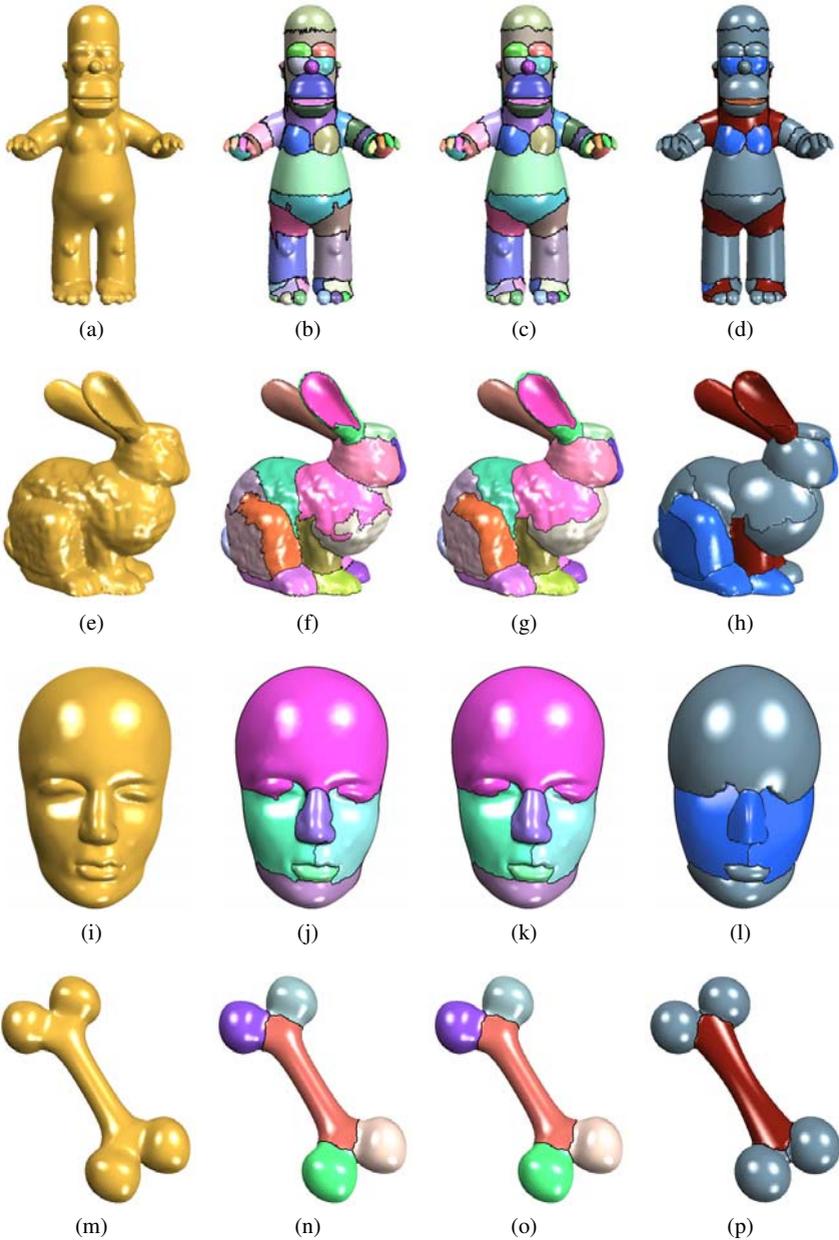
Fig. 9. Four free-form models. From left to right, the four figures on each row are the original model, final partitioned result, result after boundary smoothing, and result after vertex projection. (a)-(d) Homer (40K triangle faces): Approximated by 61 quadric proxies ($\lambda = 6.92$); (e)-(h) Bunny (40K triangle faces): Approximated by 28 quadric proxies ($\lambda = 5.49$); (i)-(l) Mask (62K triangle faces): Approximated by 6 quadric proxies ($\lambda = 4.06$); (m)-(p) Bone (30K triangle faces): Approximated by 5 quadric proxies ($\lambda = 6.2$).

Table 1. Timing Statistics

Mesh	No. of faces	No. of proxies	Lloyd iteration (sec.)	Post-processing (sec.)
Fandisk	13K	22	11	0.017
Tesa	22K	12	18	0.042
CSG	21K	10	16	0.031
Chess	24K	12	15	0.100
Bunny	40K	28	103	0.352
Homer	40K	61	148	0.393
Mask	62K	6	203	0.115
Bone	30K	5	14	0.067

of the projected results is given in Fig. 1(e). All the examples were run on a PC with a Xeon (TM) 2.66GHz CPU. Table 1 gives the running time and other statistics of all the examples.

It can be seen that our method works well for free-form geometry (cf. Fig. 9) as well as for CAD models or CSG objects with salient features (Fig. 5, 6, 7). The RMS Hausdorff errors (divided by the bounding box diagonal) by our method and the methods in [1] and [2] are presented in Fig. 8. It can be seen that the new method gives a more compact or more accurate approximation.

Acknowledgments

References

1. D. Cohen-Steiner, P. Alliez, and M. Desbrun. Variational shape approximation. *ACM Transactions on Graphics*, 23(3):905–914, 2004.
2. J.H. Wu and L. Kobbelt. Structure recovery via hybrid variational surface approximation. *Computer Graphics Forum*, 24(3):277–284, 2005.
3. S. Lloyd. Least square quantization in PCM. *IEEE Trans. Inform Theory*, 28:129–137, 1982.
4. P. D. Simari and K. Singh. Extraction and remeshing of ellipsoidal representations from mesh data. In *Graphics Interface*, pages 161–168, 2005.
5. D. Julius, V. Kraevoy, and A. Sheffer. D-charts: Quasi-developable mesh segmentation. *Computer Graphics Forum*, 24(3):981–990, 2005.
6. S. Katz and A. Tal. Hierarchical mesh decomposition using fuzzy clustering and cuts. *ACM Transactions on Graphics*, 22(3):954–961, 2003.
7. M. Attene, B. Falcidieno, and M. Spagnuolo. Hierarchical mesh segmentation based on fitting primitives. *The Visual Computer*, 22(3):181–193, 2006.
8. P.J. Besl and R.C. Jain. Segmentation through variable-order surface fitting. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 10(2):167–192, March 1988.
9. A.F. Fitzgibbon, D.W. Eggert, and R.B. Fisher. High-level CAD model acquisition from range images. *Computer-Aided Design*, 29(4):321–330, 1997.

10. A.P. Mangan and R.T. Whitaker. Partitioning 3D surface meshes using watershed segmentation. *IEEE Trans. on Visualization and Computer Graphics*, 5(4):308–321, 1999.

11. P.V. Sander, Z.J. Wood, S.J. Gortler, J. Snyder, and H. Hoppe. Multi-chart geometry images. In *Proceedings of Eurographics Symposium on Geometry 2003*, pages 146–155, 2003.

12. Y. Lee, S. Lee, A. Shamir, D. Cohen-Or, and H.-P. Seidel. Mesh scissoring with minima rule and part salience. *Computer Aided Geometric Design*, 22(5):444–465, 2005.

13. M. Marinov and L. Kobbelt. Automatic generation of structure preserving multiresolution models. *Computer Graphics Forum*, 24(3):479–486, 2005.

14. R. Liu and H. Zhang. Segmentation of 3D meshes through spectral clustering. In *Proceedings of Pacific Graphics 2004*, pages 298–305, 2004.

15. S. Katz, G. Leifman, and A. Tal. Mesh segmentation using feature point and core extraction. *The Visual Computer*, 21(8-10):649–658, October 2005.

16. G. Lavoué, F. Dupont, and A. Baskurt. A new CAD mesh segmentation method, based on curvature tensor analysis. *Computer-Aided Design*, 37(10):975–987, 2005.

17. A. Shamir. A formulation of boundary mesh segmentation. In *Proceedings of the second International Symposium on 3DPVT*, pages 82–89, 2004.

18. G. Taubin. An improved algorithm for algebraic curve and surface fitting. In *International Conference on Computer Vision*, pages 658–665, 1993.

19. S.J. Ahn, W. Rauh, H.S. Cho, and H.J. Warnecke. Orthogonal distance fitting of implicit curves and surfaces. *IEEE Tran. on Pattern Analysis and Machine Interlligence*, 24(5):620–638, 2002.

20. G. Taubin. Estimation of planar curves, surfaces and nonplanar space curves defined by implicit equations with applications to edge and range image segmentation. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 13:1115–1138, 1991.

21. Y. Li, J. Sun, C.K. Tang, and H.Y. Shum. Lazy snapping. *ACM Transaction on Graphics*, 23(3):303–308, Aug 2004.

22. Y. Boykov and V. Kolmogorov. An experimental comparison of min-cut/max-flow algorithms for energy minimization in vision. *IEEE Tran. on Pattern Analysis and Machine Interlligence*, 26(9):1124–1137, 2004.

23. Y. Liu, H. Pottmann, and W. Wang. Constrained 3D shape reconstruction using a combination of surface fitting and registration. *Computer-Aided Design (to appear)*, 2006.