Feature Edge Extraction from Mesh Model based on Wide Range Normal Evaluation and Modified Watershed Method

Tomohiro Mizoguchi[1] Hiroaki Date[1] Satoshi Kanai[1] Takeshi Kishinami[1]

[1]Graduate School of Information Science and Technology, Hokkaido University, Sapporo, Japan

Abstract

Recently, a technique which automatically generates solid models from mesh models is required in the field of reverse engineering for more efficient product development. In this paper, we propose a new feature edge extraction method from triangular mesh models based on wide range normal evaluation and segmentation using the modified Watershed method. Our feature edge extraction method is robust for mesh models with scanning noise and different mesh connectivity, and it enables recognition of the region bounded by extracted feature edges as a face. It also enables extraction of boundary edges of the fillet surface as feature edges, which was difficult using previous methods. These characteristics are effective for automatic solid model generation. Moreover we implemented functions for selecting and collecting the required feature edges extracted at different parameter settings, and transferring the resulting feature edges into a 3D CAD system.

Keywords: watershed method, feature edges, wide range normal evaluation, triangular meshes, reverse engineering

1. Introduction

Recently, mesh models can be easily obtained by several methods such as reverse engineering (RE) and searching a 3D mesh database. These mesh models are widely used at the style design stage of product development. But at the detail design stage solid models are indispensable, therefore a technique which automatically generates solid models from mesh models is required for more efficient product development.

Many studies have reported on the technique for automatic solid model generation [1,2], and most of them began with feature edge extraction (or segmentation) and surface fitting using these feature edges follows. But there are still difficulties in previous feature edges extraction methods. We define feature edges as the boundary edges of a region consisting of a set of connected triangles which should be recognized as a face. To generate a solid model automatically, three requirements should be considered for feature edges extraction method:

(1) to extract feature edges as robustly as possible from mesh models with scanning noise and different mesh connectivity
(2) to extract the boundary edges of the fillet surface included in most industrial products as feature edges
(3) to extract completely looped feature edges so as to recognize the region bounded by feature edges as a face

There has been much research on feature edge extraction methods, and they are broadly divided into two types: one is to extract feature edges directly from mesh models, and the other one is first to segment mesh models into several regions and then to extract the boundary edges of each region as feature edges.

The methods classified as the first type [3,4] have one problem; they cannot always extract completely looped feature edges, and thus do not satisfy the third requirement.

In the methods classified as the second type, first the feature value of each vertex is calculated, secondly mesh models are segmented into several regions using these feature values and finally the boundary edges of each region are extracted as feature edges. As for feature values calculation in [5], the principal curvature using 1-ring neighborhood is used as the feature value, but it cannot robustly calculate feature values of mesh models with scanning noise. Therefore it does not satisfy the first requirement. On the other hand, in [6], the edge strength of each vertex based on the surface normal variation within a wide range neighborhood is used as the feature value, and this enables robust calculation of feature values of mesh models with scanning noise. Segmentation methods are roughly divided into two types: one is vertex-based and the other is triangle-based. Vertex-based methods [6,7,8] always generate unsegmented regions between

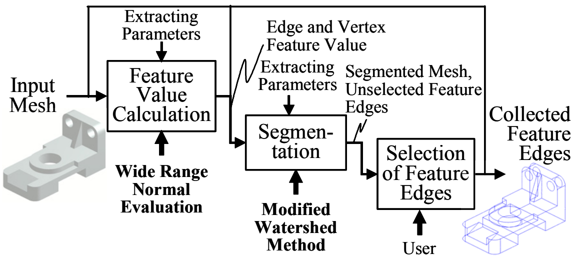


Fig.1 : Algorithm overview

regions and feature edges (boundary edges of such regions) cannot be shared between regions. On the other hand, triangle-based methods [9,10] generate no unsegmented regions and feature edges can be shared.

As for the boundary edges of the fillet surface, none of the methods mentioned above can extract them, and they therefore do not satisfy the second requirement.

In this paper, we propose a new feature edges extraction method that can satisfy all three requirements. Our method is based on wide range normal evaluation and triangle-based segmentation using the modified Watershed method. Fig.1 shows an overview of our feature edge extraction algorithm. In our method, the first edge and vertex feature values are calculated by evaluating a set of normal vectors of triangles in the wide range which is defined based on euclidean distance around each edge. Secondly, mesh models are segmented into several regions consisting of a set of connected triangles using edge and vertex feature values and the modified Watershed method. And as a result, the boundary edges of each region can be extracted as feature edges. Finally, a user selects the required feature edges from the resulting edges. By repeating these three steps at different parameter settings for extraction, users can collect a set of required feature edges.

2. Wide Range Normal Evaluation

2.1. Edge feature value calculation

In our method, wide range normal evaluation is used for feature value calculation. We define wide range as the closed space around each edge defined by the euclidean distance. This enables robust calculation of edge and vertex feature values of mesh models with scanning noise and different mesh connectivity.

To calculate edge feature values by our method, first the sphere B whose center is the mid-point of edge e and radius r is defined. Then this sphere B is decomposed into two hemispheres B_R and B_L using the plane defined by edge e and $\bar{\mathbf{n}}$, which is the average of the normal vectors of triangles included in B (Fig.2). Secondly, the sums of normal vectors of triangle $\mathbf{n}_{B_R}$

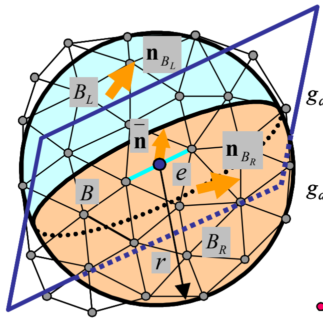


Fig.2 : Edge feature value calculation

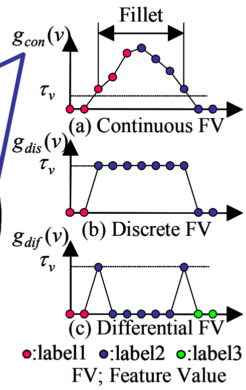


Fig.3 : Vertex feature value calculation

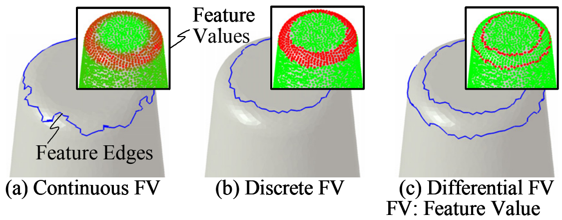


Fig.4 : Comparison of extracted feature edges

and $\mathbf{n}_{B_L}$ are calculated by the equation $\mathbf{n}_{B_R} = \sum_{f \in f_{B_R}} \mathbf{n}_f \cdot area(f)$, where $\mathbf{n}_f$ is the face normal of triangle f, $area(f)$ the area of triangle f, and f_{B_R} a set of triangle whose barycenter is included in B_R (the same for B_L). Finally, edge feature value $\theta_e(r)$ of an edge e is calculated as the angle between $\mathbf{n}_{B_R}$ and $\mathbf{n}_{B_L}$.

2.2. Vertex feature value calculation

Vertex feature values are calculated using the edge feature values. First, the *continuous feature value* of each vertex v is calculated by $g_{con}(v) = \max_{e \in e^*(v)} \theta_e(r)$, where $e^*(v)$ is a set of edges sharing v. To automatically generate a solid model using extracted feature edges, the boundary edges of the fillet surface need to be extracted so as to fit with proper surface geometry. But $g_{con}(v)$ of vertices varies in a fillet surface as shown in Fig.3(a). Using this $g_{con}(v)$, segmenting mesh models based on the modified Watershed method (mentioned in sec.3) divide a fillet surface into two different regions, and the unwanted center line of the fillet surface is extracted as shown in Fig.4(a). Therefore the vertex *discrete feature value* $g_{dis}(v)$ is defined by thresholding $g_{con}(v)$ using threshold τ_v, and this enables to set vertex feature values in a fillet surface constant as shown in Fig.3(b). But even using this $g_{dis}(v)$, a fillet surface is recognized as the same region as the one which is smoothly connected to the fillet surface, and only one boundary edge out of two is extracted as shown in Fig.4(b). Therefore the vertex *differential feature value* $g_{dif}(v)$ is defined by eq.(1) as the difference of $g_{dis}(v)$ between the vertex v and a set of 1-ring vertices v^* as shown in Fig.3(c).

$$g_{dif}(v) = \begin{cases} g_{dis}(v) - \min_{v' \in v^*} g_{dis}(v') \\ \quad (g_{dis}(v) \geq \min_{v' \in v^*} g_{dis}(v')) \\ 0 \qquad (otherwise) \end{cases} \tag{1}$$

Using this $g_{dif}(v)$, both boundary edges of the fillet surface are extracted properly, as shown in Fig.4(c). In our method, $g_{dif}(v)$ is used only for feature (boundary) edge extraction of the fillet surfaces, and $g_{con}(v)$ for feature edge extraction except for fillet surfaces.

3. Modified Watershed Method

Mangan et al.[7] extended the Watershed method which was used for 2D images segmentation into the 3D mesh model segmentation. Mangan's Watershed method is composed of three steps as follows;

(1) **Initial labeling:** Local minima vertices whose feature values are lower than those of all of their neighboring vertices are found and each region label is allocated to them. Then these vertices become seed regions.

(2) **Descend:** Other vertices without region labels are allowed to descend until they hit a vertex with a region label, and its region label is allocated to them. Then seed regions are grown and vertex-based regions are generated.

(3) **Region merging:** Regions whose Watershed depth and area are below thresholds τ_d and τ_n are merged. Then small regions are merged and become a part of larger regions.

This Watershed method has two problems for feature edge extraction. One is that it is a vertex-based method, therefore it generates unsegmented regions between regions, consisting of a set of triangles whose three vertices have different region labels, and the boundary edges of each region cannot be shared. The other problem is that this method cannot segment mesh models properly if there are not enough vertices in one region (for example, if one rectangle is composed of two right triangles). To solve these problems, we propose a modified Watershed method based on Razdan's method[9]. Our method has some modifications to realize a more robust method of feature edge extraction.

Our modified Watershed method is composed of five steps as follows.

(1) **Vertex insertion:** The edges whose edge feature values $\theta_e(r)$ are lager than threshold τ_θ are identified as *Basic Feature Edges* (BFE), and the two vertices of BFE as *Basic Feature Vertices* (BFV). Then vertices and edges are inserted according to the inserting rule (Fig.5). This is done in order to recognize the region which does not have enough vertices in it.

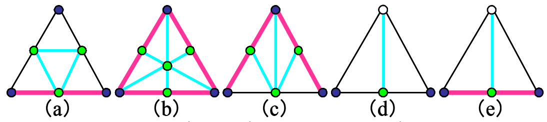


Fig.5 : Vertices and edges insertion rules

Table1 : Triangle region label allocation rules

	Vertex Labels	Allocated Triangular Label
case1	All vertices have the same label	The label of its vertices
case2	One vertex has a single label	The single label
case3	Multiple labels and only one common label	The common label
case4	All edges are BFE	New label
case5	Multiple labels and multiple common labels	The other label with the triangle that shares the BFE
case6	otherwise	The label of the vertex that has the smallest vertex value

(2) **Vertex feature value calculation:** Vertex feature values of the inserted vertices are allocated as follows; the maximum feature values out of all feature values of the original vertices are allocated for inserted vertices on BFE, the minimum feature value out of them for the inserted vertices on the center of the triangle, and the original edge feature values for other vertices.

(3) **Watershed method:** One region label for each vertex (including inserted vertices) is allocated by applying the Watershed method to the mesh model which has inserted vertices and edges.

(4) **Vertex deletion:** The region labels of 1-ring vertices are attached to BFV's original region label so that it has multiple region labels, and then the inserted vertices are deleted from the mesh models and the original mesh model is restored.

(5) **Triangle region label allocation:** One region label is allocated to each triangle according to the triangular region label allocating rule (Table1). This enables generation of regions consisting of connected triangles, which have the same region labels, and feature edges can be extracted as completely looped boundary edges of these regions.

This modified Watershed method can solve the two problems of Mangan's Watershed method. But extraction of all the required feature edges is difficult to achieve by applying this method once at one parameter setting. Five extraction parameters are used in our method; radius r of the sphere for defining the evaluating area for feature value calculation, the threshold τ_v for thresholding the vertex continuous feature value, the threshold τ_θ for identifying BFE, and the threshold τ_d and τ_n for defining the smallest

Watershed depth and the smallest area of each region. Then a user repeats this method at different parameter settings, and selects and collects the required feature edges extracted at different parameter settings.

4. Results

Fig.6 shows the results of feature edges extraction from mesh models with scanning noise (710,000 triangles) generated by 3D scanning of hand-made design mock-up. This shows our method robustly extract boundary edges of fillet surface as feature edges from mesh models with scanning noise by setting parameters properly. The running time was 140s (80s for feature values calculation and 60s for segmentation and feature edge extraction) for automatic extraction at one parameter setting on the PC (Xeon 2.4GHz, 2GB RAM).

Fig.7 shows the results of collection of selected feature edges by user interaction. The input mesh model (60,000 triangles) was generated by FEM meshing of a solid model. Fig.7(a)-(c) shows selected feature edges at each three different parameter settings; thin black edges are all extracted edges at one parameter setting, and thick blue edges are interactively selected edges. And Fig.7(d) shows collected feature edges which are considered to be effective for automatic solid model generation.

5. Conclusion

In this paper, we proposed a new feature edge extraction method from triangular meshes based on wide range normal evaluation and triangular-based segmentation using the modified Watershed method. Experimental results show that our method enables robust extraction of feature edges from mesh models with scanning noise and different mesh connectivity, extraction of boundary edges of fillet surface as feature edges, and extraction of completely looped feature edges so as to recognize the region bounded by feature edges as a face. And we also implemented interactive extraction functions for helping solid model generation. Our future work includes the automatic estimation of surface geometry which should be fitted to each region bounded by extracted feature edges.

References

[1] W.B.Thompson et al. Feature-Based Reverse Engineering of Mechanical Parts. *IEEE Trans. on Robotics and Automation*, 15(1), 1-10, 1999.

[2] P.Benko et al. Algorithms for reverse engineering boundary representation models. *Computer-Aided Design*, 33(11), 839-851, 2001.

[3] T.Watanabe et al. Extraction Method of Feature Edges of a Shape Modeled by Using Arbitrary Triangular Meshes. *IEICE Trans. on Information and Systems*, J83 D-2(5), 1344-1352, 2000

[4] Y.Ohtake et al. Ridge-Valley Lines on Meshes via Implicit Surface Fitting. *ACM Trans. on Graphics*, 23(3), 609-612, 2004.

[5] M.Meyer et al. Discrete Differential-Geometry Operators for Triangulated 2-Manifolds. *Proc. of Visualization and Mathematics*, 3, 34-57, 2002.

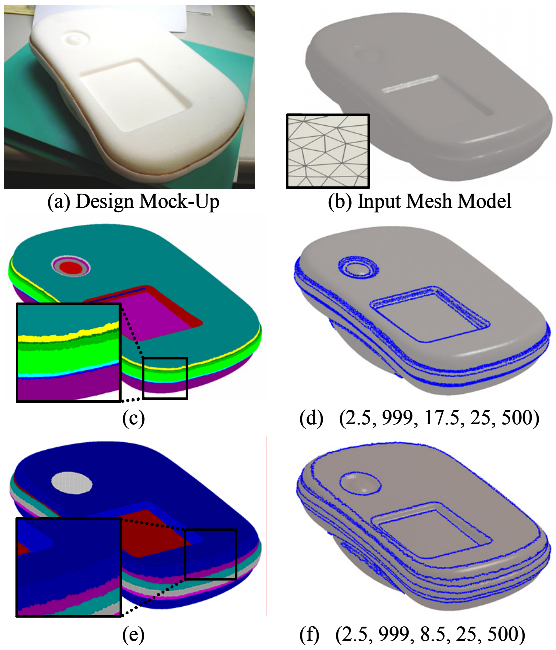
[6] Y.Sun et al. Triangle mesh-based edge detection and its application to surface segmentation and adaptive surface smoothing. *Proc. IEEE ICIP 2002*, 825-828, 2002.

[7] A.Mangan et al. Partitioning 3D Surface Meshes Using Watershed Segmentation. *IEEE Trans. on visualization and computer graphics*, 5(4), 308-321, 1999.

[8] M.Vieira et al. Segmentation of Noisy Laser-Scanner Generated Meshes with Piecewise Polynomial Approximations. *Proc. of DETC'04*, 2004.

[9] A.Razdan et al. A hybrid approach to feature segmentation of triangle meshes. *Computer-Aided Design* 35(9), 783-789, 2003.

[10] S.Katz et al. Hierarchical mesh decomposition using fuzzy clustering and cuts. *ACM Trans. on Graphics*, 22(3), 954-961, 2003.



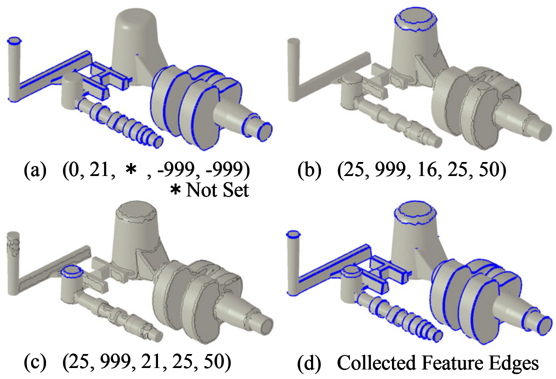
(a) Design Mock-Up (b) Input Mesh Model

(c) (d) (2.5, 999, 17.5, 25, 500)

(e) (f) (2.5, 999, 8.5, 25, 500)

(Parameters for Extraction $r, \tau_\theta, \tau_v, \tau_d, \tau_n$)

Fig.6 : Results of feature edges extraction



(a) (0, 21, ＊, -999, -999) (b) (25, 999, 16, 25, 50)

＊Not Set

(c) (25, 999, 21, 25, 50) (d) Collected Feature Edges

(Parameters for Extraction $r, \tau_\theta, \tau_v, \tau_d, \tau_n$)

Fig.7 : Results of feature edges collection