

Segmentation of Scanned Mesh into Analytic Surfaces Based on Robust Curvature Estimation and Region Growing

Tomohiro Mizoguchi, Hiroaki Date, Satoshi Kanai, and Takeshi Kishinami

Graduate School of Information Science and Technology, Hokkaido University,
Kita 14, Nishi 9, Kita-ku, Sapporo, 060-0814, Japan
tmizoguchi@minf.coin.eng.hokudai.ac.jp,
{hdate, kanai, kishinami}@ssi.ist.hokudai.ac.jp

Abstract. For effective application of laser or X-ray CT scanned mesh models in design, analysis, and inspection etc, it is preferable that they are segmented into desirable regions as a pre-processing. Engineering parts are commonly covered with analytic surfaces, such as planes, cylinders, spheres, cones, and tori. Therefore, the portions of the part's boundary where each can be represented by a type of analytic surface have to be extracted as regions from the mesh model. In this paper, we propose a new mesh segmentation method for this purpose. We use the mesh curvature estimation with sharp edge recognition, and the non-iterative region growing to extract the regions. The proposed mesh curvature estimation is robust for measurement noise. Moreover, our proposed region growing enables to find more accurate boundaries of underlying surfaces, and to classify extracted analytic surfaces into higher-level classes of surfaces: fillet surface, linear extrusion surface and surface of revolution than those in the existing methods.

1 Introduction

3D laser and X-ray CT scanning systems are widely used in the field of reverse engineering to acquire scanned data of real-world objects. To use the acquired scanned data in today's digital engineering, it is easily converted into a 3D mesh model by a surface reconstruction algorithm such as marching cubes [1]. When we utilize 3D scanned mesh model for repairing, replication, analysis, or inspection of engineering parts, we need to efficiently segment the mesh model into desirable regions depending on its applications.

The surfaces of engineering parts mainly consist of a set of analytic surfaces, such as planes, cylinders, spheres, cones, and tori. Therefore, we need to extract regions each of which can be approximated by a simple analytic surface from a mesh model. However, few methods have been proposed to extract analytic surfaces from a mesh model. Moreover, in these methods, the accuracies of extracting regions from noisy mesh models and the range of extracted analytic surface classes were not necessarily sufficient from the aspect of practical engineering use.

The purpose of this research is to propose a new method that segments a scanned mesh model into regions each of which can be approximated by a simple analytic

surface. In this paper, we only deal with triangular mesh models whose surfaces are completely composed of planes, cylinders, spheres, cones, and tori.

Our algorithm is composed of three steps. The first step accurately estimates mesh principal curvatures based on a modified method of Razdan's [2]. It allows robust estimation for a noisy scanned mesh and ensures more accurate estimation even around sharp edges where the previous methods generated large estimation error (section 3). The second step extracts analytic surfaces based on the modified version of Vieira's region growing algorithm [3]. Our curvature estimation and limiting the fitting surface only to the analytic enable to initially create large seed regions in the region growing. This also enables non-iterative region growing and the efficient linear LSM in analytic surface fitting (section 4). The final step classifies extracted analytic surfaces into higher-level classes of surfaces than those in the existing methods [4][5][6]: fillet surface, linear extrusion surface, and surface of revolution.

2 Related Works

Mesh segmentation is a technique that segments a mesh model into desirable regions depending on applications, and many methods have been proposed for this segmentation in computer graphics(CG)[7][8][9] and engineering field. These works in CG are aiming at decomposing mesh surfaces into visually meaningful sub-meshes. On the other hand, the goal of the segmentation in an engineering field aims at decomposing the mesh surface into functionally meaningful surfaces. Therefore segmentation in CG cannot be directly applied to the engineering purpose. The mesh segmentation in the engineering field is roughly divided into following three groups:

The first group is to extract regions separated by sharp edges on a mesh model. In this group, a *watershed-based* approach has been well studied [10][11][12]. However they cannot extract regions separated by smooth edges (i.e., a region consisting of a plane smoothly connected to a cylinder), and therefore cannot identify the surface geometry of each segmented region.

The second group is to extract regions each of which can be approximated by a simple free form surface. In this group, a *region growing* approach [3][13][14] has been well studied. However the method did not focus on extracting regions approximated by analytic surfaces and their geometric parameters.

The last group is to extract regions each of which can be approximated by a simple analytic surface. Gelfand *et al.*[4] proposed a method based on eigenvalue analysis of a mesh model. Wu *et al.*[5] proposed a method based on Lloyd's clustering algorithm. However, in their method, the range of extracted analytic surface classes was not necessarily sufficient for engineering applications. Benkő *et al.*[6] proposed the *direct segmentation* for reverse engineering of the engineering part. Their algorithm segments a mesh model into regions each of which can be approximated by simple analytic surfaces (planes, cylinders, spheres, cones, and tori), linear extrusion surfaces, and the surface of revolutions. However this algorithm results in poor segmentation near boundaries of surfaces where the indicators may not be properly estimated. And they applied their segmentation only for a mesh model with very simple geometry.

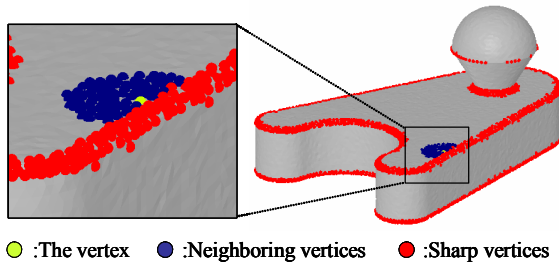


Fig. 1. Curvature estimation with sharp edges recognition

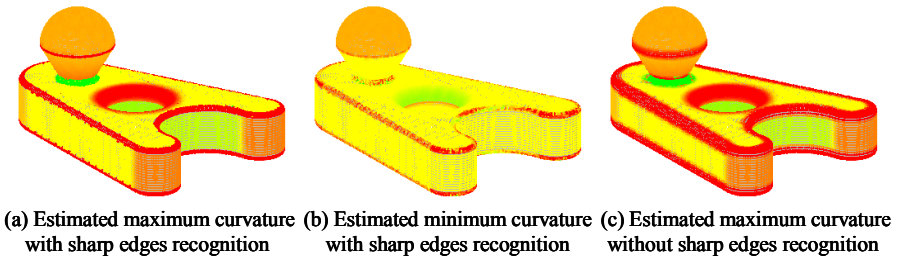


Fig. 2. Results of curvature estimation

3 Robust Mesh Curvatures Estimation by Recognizing Sharp Edges

To estimate mesh curvatures on a noisy meshes, Razdan proposed the method based on a local biquadratic Bézier surface fitting [2]. This method locally fits the surface for a mesh vertex and a set of vertices included in its 2-ring, and estimates mesh curvatures at the vertex from the fitted surface. We simply modify this method, and fits the surface for each vertex v_i and a set of vertices directly connected to v_i within those which satisfy the condition of eq.(1),

$$\| \mathbf{v}_j - \mathbf{v}_i \| < W \cdot l_{i,avg} \tag{1}$$

where $l_{i,avg}$ is an average length of edge connecting to the vertex v_i , and W is a parameter to specify the size of surface-fit. Our algorithm analyses the fitted surface and estimates two principal curvatures by the method [2] with the parameter $\alpha=0.9$. Then we define the one whose absolute value is the larger as the maximum principal curvature $\kappa_{i,max}$, and the other one as the minimum principal curvature $\kappa_{i,min}$.

To robustly estimate principal curvatures on noisy scanned meshes, a parameter W should be larger. However, a larger parameter setting results in poor curvature estimation around sharp edges. To solve the problem, we propose the following “two-pass” curvature estimations.

In the first pass, principal curvatures are estimated by fitting the relatively small surface on the mesh with $W=2$. If the maximum curvature $\kappa_{i,\max}$ of the vertex is larger than the user specified parameter th_{se} , the vertex is classified as a *sharp vertex* as shown in Fig.1. This parameter setting is as the same as Vieira's method [3].

In the second pass, principal curvatures are re-calculated for non-sharp vertices with $W=5$. In this step, sharp vertices and neighboring vertices beyond them are not included for the Bézier surface fitting as shown in Fig.1. This results in better accuracy of principal curvatures of vertices near sharp edges as shown in Fig.2.

Moreover in our method, the normal vector \mathbf{n}'_i for the vertex v_i is also re-calculated as the normal of the fitted Bézier surface. This normal vector is better in accuracy than the one \mathbf{n}_i calculated using connected triangles for noisy mesh models. The normal vectors \mathbf{n}_i are used for region growing described in section 5.2, and the re-calculated normal vectors \mathbf{n}'_i are used for the analytic surface fitting described in section 5.1.

4 Seed Region Creation

Next, seed regions are created on the surface of a mesh by classifying estimated principal curvatures. In this paper, a seed region means a set of topologically connected vertices that are supposed to be on a certain analytic surface. Our seed region creation algorithm is composed of following four steps.

Step1: Allocation of labels for planes, cylinders/cones, and spheres

A label for plane, cylinder/cone, and sphere that discriminates on which surface the vertex v_i lies is allocated to it. Our method allocates the labels for non-sharp vertices classified in sec.3 according to eq.(2) by evaluating two principal curvatures. Three parameters $\varepsilon_1, \varepsilon_2, \varepsilon_3$ are used in eq.(2). Ideally, $\varepsilon_1, \varepsilon_2, \varepsilon_3$ are zero, however they may not be zero due to scanning noise. We found that $\varepsilon_1 = \varepsilon_2 = \varepsilon_3 = 0.01$ roughly provided good results for mesh models both with artificially-added and real-scanning noise.

$$label(v_i) = \begin{cases} 1 & (plane) & \text{if } (|\kappa_{i,\max}| < \varepsilon_1) \\ 2 & (cylinder, cone) & \text{else if } (|\kappa_{i,\min}| < \varepsilon_2) \\ 3 & (sphere) & \text{else if } (|\kappa_{i,\max} - \kappa_{i,\min}| < \varepsilon_3 \\ & & \text{and } \kappa_{i,\max} \cdot \kappa_{i,\min} > 0) \\ 0 & (others) & \text{otherwise} \end{cases} \quad (2)$$

Step2: Classification of cylinders/cones and smoothly connected cylinders

Step 1 allocates the label 2 for both cylinders and cones. In this step, these cylinders and cones are discriminated. The smoothly connected cylinders are also separated into different single cylinders. To achieve this, a similarity value of the principal maximum curvature $f(v_i)$ is calculated for each vertex v_i according to eq.(3).

Table 1. Thresholds for seed region creation

surface type	threshold th_{seed}
plane	2
cylinder	4
sphere	5
cone	4
torus	7

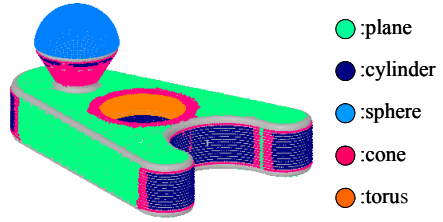


Fig. 3. Result of seed region creation

$$f(v_i) = \frac{1}{n} \sum_{j \in N^{**}(i)} |(\kappa_{i,max} - \kappa_{j,max}) / \kappa_{i,max}| \tag{3}$$

where $N^{**}(i)$ is a set of vertices in 2-ring of the vertex v_i . Ideally, $f(v_i)$ is zero in cylinders and positive value in cones. Therefore if $f(v_i)$ is larger than the threshold th_{cyl_cone} , the vertex is assumed to belong to a cone, and $label(v_i)$ is changed to 4 and if $f(v_i)$ is smaller than the threshold, $label(v_i)$ is preserved. For our implementation, $th_{cyl_cone} = 0.01 - 0.4$ provides a good result.

Step3: Allocation of labels for tori

Previous two steps allocate labels for plane, cylinder, sphere, and cone, therefore most of vertices with label 0 are assumed to lie on tori. To allocate torus labels to such vertices, principal curvatures of vertices that have label 0 are evaluated.

A torus is the excursion surface where a sphere is rotated along an axis. Therefore one of the principal curvatures corresponds to the constant curvature of the radius r of the sphere. We use this property and create a histogram of discretized principal curvatures for a set of vertices that have the label 0. If the number of vertices that has a particular discretized curvature value is larger than the threshold th_{torus} , label 5 is allocated to their vertices. In our implementation, 0.01 for the step of principle curvature, and 0.1-0.5% of the number of all vertices for th_{torus} provided good results for most mesh models.

Step4: Creation of seed regions

Finally, a seed region is created as a set of topologically connected vertices with the same label that has the number of vertices larger than the threshold th_{seed} shown in table1. In our implementation, th_{seed} corresponds to the minimum number of vertices which enables the least square analytic surface fitting that is proposed in this paper and is described in section 5.1. Fig.3 shows the results of seed region creation.

5 Analytic Surface Extraction

5.1 Analytic Surfaces Fitting to Seed Regions

In this paper, we propose the following efficient analytic surface fitting where we only need to solve the linear least squares problems to find fitted analytic surfaces

instead of the non-linear one by utilizing pre-computed normal vectors \mathbf{n}'_i . Our method is less accurate than previous non-linear methods [15][16][17], but faster than them, and provides a practically enough results for analytic surface extraction.

Plane fitting: A plane is defined by its unit normal vector $\mathbf{n} = (n_x, n_y, n_z)$ and a distance d from an origin. Our method calculates \mathbf{n} as the normalized average vector of vertex normals \mathbf{n}'_i in a seed region of the plane. Then the distance d is calculated using linear least squares sense.

Cylinder fitting: A cylinder is defined by its unit axis direction vector $\mathbf{d} = (d_x, d_y, d_z)$, radius r , and an arbitrary point on the axis $\mathbf{p} = (p_x, p_y, p_z)$. First, all of vertex normals \mathbf{n}'_i in a seed region are mapped onto a Gaussian sphere. Then a least square plane is fitted so that the plane passes through the end points of \mathbf{n}'_i in the sphere. The axis direction is derived as a unit normal vector of this plane. Next, all vertices in a seed region are projected onto the plane whose normal vector is \mathbf{d} . Then, a circle is fitted to these projected points on the plane in the least squares sense, and the radius r is calculated as the radius of the fitted circle. Finally, the center of the circle is also calculated on the projected plane, and it can be easily transformed to \mathbf{p} .

Sphere fitting: A sphere is defined by its center $\mathbf{c} = (c_x, c_y, c_z)$ and radius r . Our method solves a linear least squares problem to find the coefficients (A, B, C, D) defining the sphere of the implicit form $x^2 + y^2 + z^2 + Ax + By + Cz + D = 0$. They are easily converted into the center \mathbf{c} and radius r .

Cone fitting: A cone is defined by its unit axis direction vector $\mathbf{d} = (d_x, d_y, d_z)$, apex $\mathbf{a} = (a_x, a_y, a_z)$, and vertical angle θ . The unit axis direction vector \mathbf{d} is calculated using the same method as for a cylinder. An apex \mathbf{a} is given from the condition that a vector passing through \mathbf{a} and \mathbf{v}_i is orthogonal to a normal vector of the vertex \mathbf{n}'_i . This is obtained by finding a least squares solution of \mathbf{a} in $\mathbf{n}'_i \cdot (\mathbf{a} - \mathbf{v}_i) = 0$. An apex θ is calculated as the average of angles between \mathbf{d} and a vertex normal \mathbf{n}'_i .

Torus Fitting: A torus is defined by its unit axis direction vector $\mathbf{d} = (d_x, d_y, d_z)$, its center $\mathbf{c} = (c_x, c_y, c_z)$, the radius of its body r , and the radius of the centerline of the torus body R . First, to calculate \mathbf{d} and an arbitrary point \mathbf{p} on the axis, we use the same method as Kós *et al.*[15]. Kós first calculated the initial estimates of \mathbf{d} and \mathbf{p} using the generalized eigen analysis, and then the precise solutions of them are calculated by the iterative method from these initial estimates. We first use Kós's initial estimates as our final solutions of \mathbf{d} and \mathbf{p} for fitting torus. Next, all vertices in a seed region are rotated around the calculated axis so as to be placed onto a plane which includes the calculated axis. Then a circle is fitted to the points on the plane in the least squares sense. The radius r is calculated as the radius of that circle. The center \mathbf{c} is calculated according to the condition that the vector toward the center of the torus \mathbf{c} from the center of the circle is orthogonal to the torus axis. The radius R is also calculated as the length between the center of the torus and the center of the circle.

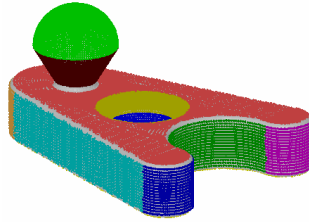


Fig. 4. Results of extracted analytic surfaces using the region growing

5.2 Extraction of Analytic Surfaces Based on Region Growing

Next, our region growing method makes a set of vertices topologically connected to the seed region added to the original seed region if the vertex lies on the fitted surface within a specified tolerance. Therefore our algorithm first calculates the positional error between the mesh vertex v_i and the point $p(v_i)$ on the analytic surface which is the projection of v_i to the surface along \mathbf{n}_i , and the directional error of the normal vectors between them. If the vertex is adjacent to the seed region and satisfies the compatibility in eq.(4) and eq.(5), it is added to the seed region.

$$|\mathbf{v}_i - \mathbf{p}(v_i)| < th_{pos} \cdot l_{avg} \tag{4}$$

$$|\cos^{-1}(\mathbf{n}_i - \mathbf{n}(p(v_i)))| < th_{norm} \tag{5}$$

where l_{avg} is an average length of all mesh edges. th_{pos} and th_{norm} are the thresholds of positional and directional errors, and we found in the experiments that $th_{pos} = 0.5$ and $th_{norm} = 8.0[\text{deg}]$ provided good results for mesh models measured from CT scanning.

This region growing is done for the seed region in the descending order of the number of vertices in the region. This enables to generate a small number of larger regions. If all the vertices adjacent to the region do not satisfy eq.(4) or eq.(5), the region growing stops. The region can be extracted as a set of topologically connected vertices that are approximated by a particular analytic surface. Fig.4 shows the result of analytic surfaces extraction.

6 Recognition of Fillet Surfaces, Linear Extrusion Surfaces, and Surfaces of Revolution

For the effective use of a mesh model in various mesh applications, our method recognizes fillet surfaces, linear extrusion surfaces, and surfaces of revolutions which are often included in most engineering parts from a mesh model.

Recognition of fillet surfaces

We assume that all surfaces in a mesh model are covered with analytical surfaces, and fillet surfaces are also represented by them. This assumption enables to classify fillet

	A torus-type between two surfaces	A cylinder-type between two planes	A sphere/torus-type between three cylinder-type fillets	
Recognized fillets				
Neighboring surfaces	A cylinder and a plane	A sphere and a plane	Two planes	
Geometric conditions	$n_p \parallel d_t \parallel d_c$	<ol style="list-style-type: none"> p_c lie on the torus axis $n_p \parallel d_t$ 	<ol style="list-style-type: none"> $n_{p2} \perp d_c$ and $n_{p1} \perp d_c$ $0^\circ < \theta < 180^\circ$ 	<p><u>Sphere-type:</u> only convex or concave are connected</p> <p><u>Torus-type:</u> convex and concave are connected</p>

Fig. 5. Definition of fillet surfaces

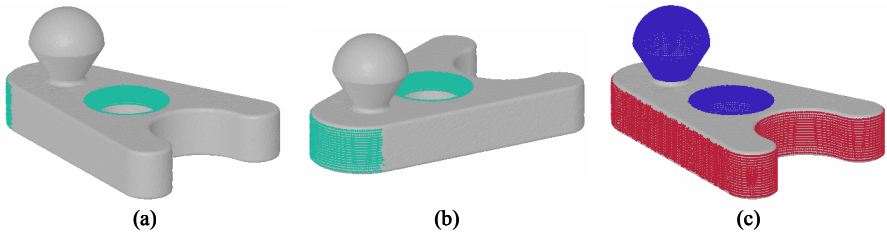


Fig. 6. (a)(b)Recognition of fillet surfaces, (c) a linear extrusion surface (red) and surfaces of revolution (blue)

surfaces into three types: cylinders, spheres, and tori [18]. These surfaces can be defined based on their geometric parameters and combinations of neighboring surfaces as shown in Fig.5. Our method recognizes the analytic surface satisfying the definition as a fillet surface. Fig.6(c) shows results of recognizing fillet surfaces.

Recognition of linear extrusion surfaces

The linear extrusion surface is composed of a combination of some planes and cylinders. These surfaces must satisfy the following three conditions: (1) a plane normal and a cylinder axis must be orthogonal, (2) normal vectors of three or more planes must be coplanar, and (3) axes of two or more cylinders must be parallel. Our method recognizes a set of topologically connected analytic surfaces satisfying the above conditions as a linear extrusion surface. Fig.6(c) shows results of recognizing a linear extrusion surface.

Recognition of surfaces of revolution

The surface of revolution is composed a combination of planes, cylinders, spheres, cone, and tori. These surfaces must satisfy the following two conditions: (1) normal vectors of planes and axis directions must be parallel, and (2) center points of spheres and tori, apexes of cones, and arbitrary points of axes of cylinders must lie in a same line with a direction parallel to their normals or axis. Our method recognizes a set of topologically connected analytic surfaces satisfying the above conditions as a surface of revolution. Fig.6(c) also shows results of recognizing a surface of revolution.

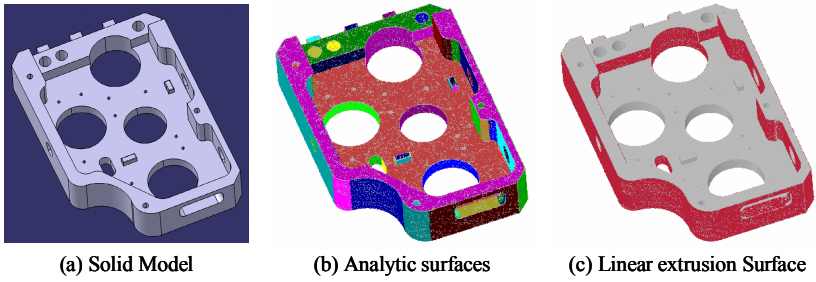


Fig. 7. Results to our mesh model for verification

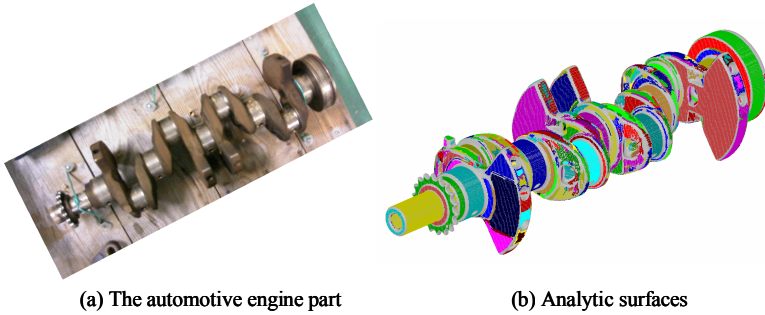


Fig. 8. Results to the mesh model created by CT scanning (974,754 tri)

7 Results

Fig.7 shows the results to a mesh model (300,000tri) for verification which was created by FEM meshing of a solid model. Then we added artificial noise on this model by moving the vertex position to its normal direction using a Gaussian distributed random value. Our method can extract regions from a noisy complex shaped model and can find accurate boundaries of underlying analytic surfaces.

Fig.8 shows the results for the CT-scanned mesh model of an automotive engine part. It shows that our method could well extract planes and cylinders with relatively large areas from the model. Especially it could extract all functionally important cylindrical regions (fitted with bearings). The model has about 1,000,000 triangles and our method could extract analytic surfaces in less than 7 minutes.

8 Conclusions and Future Works

In this paper, we proposed a new method of systematically extracting analytic surfaces from a mesh model. From the simulation and the experiment for the various mesh models, we found that our method could produce an accurate and practical geometric model consisting of a set of analytic surfaces, fillet surfaces, linear extrusion surfaces and surfaces of revolution from mesh models.

One limitation of our method is that the thresholds to extract regions cannot be easily set by users. In our research, we experimentally found the appropriate values for all thresholds described in this paper, and they can work well for various mesh models.

As for our future work, we need to impose geometric constraints among fitted surfaces (parallel, orthogonal, continuous, etc). Moreover, feature recognition such as boss, lib, slot, etc., will also be needed to use mesh models as we use feature-based solid models that are commonly used in the commercial 3D CAD systems.

Acknowledgement

This work was financially supported by a grant-in-aid of Intelligent Cluster Project (Sapporo IT Carrozzeria) founded by MEXT. We thank for Ichiro Nishigaki and Noriyuki Sadaoka in HITACHI Co.Ltd. for providing CT-scanned mesh model.

References

1. Lorensen, W.E., Harvey, E.C.: Marching cubes: A high resolution 3D surface construction algorithm. *ACM SIGGRAPH Computer Graphics*, Vol.21, No.4. (1987) 163-169
2. Razdan, A., Bae, M.S.: Curvature estimation scheme for triangle meshes using biquadratic Bézier patches. *Computer-Aided Design*, Vol.37, No.14. (2005) 1481-1491
3. Vieira, M., Shimada, K.: Surface mesh segmentation and smooth surface extraction through. *Computer-Aided Geometric Design*, Vol.22, No.8. (2005) 771-792
4. Gelfand, N., Guibas, L.J.: Shape segmentation using local slippage analysis. *Proc. of Eurographics/ACM SIGGRAPH symposium on Geometry processing*. (2004) 214-223
5. Wu, J., Kobbelt, L.: Structure Recovery via Hybrid Variational Surface Approximation. *Proc. of Eurographics*. Vol.24, No.3. (2005) 277-284
6. Benkő, P., Várady, T.: Segmentation methods for smooth point regions of conventional engineering objects. *Computer-Aided Design*, Vol.36, No.6. (2004) 511-523
7. Yamauchi, H., Gumhold, S., Zayer, R., Seidel, H.P.: Mesh Segmentation Driven by Gaussian Curvature. *Visual Computer*. Vol.21, No.8-10. (2005) 649-658
8. Katz, S., Leifman, G., Tal, A.: Mesh segmentation using feature point and core extraction. *Proc. of Pacific Graphics*. (2005) 649-658
9. Attene, M., Katz, S., Mortara, M., Patané, G., Spagnuolo, M., Tal, A.: Mesh Segmentation - A comparative study. *Proc. of Shape Modeling and Applications*. (2006)
10. Mangan, A.P., Whitaker, R.T.: Partitioning 3D Surface Meshes Using Watershed Segmentation. *IEEE Trans. on visualization and computer graphics*, Vol.5, No.4. (1999) 308-321
11. Sun, Y.D., Page, L., Paik, J. K., Koschan, A., Abidi, M.A.: Triangle mesh-based edge detection and its application to surface segmentation and adaptive surface smoothing. *Proc. of the International Conference on Image Processing*. Vol.3. (2002) 825-828
12. Razdan, A.: Hybrid approach to feature segmentation of triangle meshes. *Computer-Aided Design*, Vol.35, No.9. (2003) 783-789
13. Besl, P.J., Jain, R.C.: Segmentation through Variable-Order Surface Fitting. *IEEE Trans. on Pattern Analysis and Machine Intelligence*. Vol.10, No.2. (1988) 167-192

14. Djebali, M. Melkemi, M., Sapidis, N.: Range-Image segmentation and model reconstruction based on a fit-and-merge strategy. Proc. of ACM symposium on Solid modeling and applications.(2002) 127-138
15. Kós, G., Martin, R., Várady, T.: Methods to recover constant radius rolling blends in reverse engineering. Computer-Aided Geometric Design. Vol.17, No.2. (2000) 127-160
16. Benkő, P., Kós, G., Várady, T., Andor, L., Martin, R.: Constrained fitting in reverse engineering. Computer-Aided Geometric Design. Vol.19, No.3. (2002) 173-205
17. Marshall, D., Lukacs, G., Martin, R.: Robust Segmentation of Primitives from Range Data in the Presence of Geometric Degeneracy. IEEE Trans. on Pattern Analysis and Machine Intelligence. Vol.23, No.3. (2001) 304-314
18. Zhu, H., Menq, C.H.: B-Rep model simplification by automatic fillet/round suppressing for efficient automatic feature recognition. Computer-Aided Design. Vol.34, No.2. (2002) 109-123