3D Terrain Reconstruction Based on Contours

Zhiyi Zhang
Iwate University
Faculty of Engineering
Morioka-shi, Japan 020-8551

Kouichi Konno
Iwate University
Faculty of Engineering
Morioka-shi, Japan 020-8551

Yoshimasa Tokuyama
Tokyo Polytechnic University
Faculty of Engineering
Atsugi-shi, Japan 243-0297

Abstract

In 3D terrain reconstruction, the differences of shape and topology among contours in adjacent sections cause a difficulty as the tiling problem of branching terrain. In this paper, we present a brief and pure geometrical algorithm to reconstruct 3D terrain based on contours. The merit of our method is that the tiling rules guarantee arbitrary branching terrain can be divided into correct topology.

1. Introduction

Contour lines from topographical maps are still the most common form of elevation data for the earth's surface. However, contours maps have generally insufficient precision for disaster prevention, environment analysis, education, entertainment and other several purposes. For example, 10m$\times$10m DEM(digital elevation model) is the precisest data in Japan. When layering contours with 5m, 2m or 1m accuracy are required, we have to interpolate present contous into surface model, then cut the surface model to get contours with desirous accuracy. Therefore, a method for constructing a surface model from parallel contours is one of the most important problems in terrain application.

Traditionally, the problem of surface reconstruction from contours can be broken down into three subproblems: correspondence problem, branching problem and tiling problem [3][6][2][1][15]. The correspondence problem arises when there are two or more contours in adjacent two sections, which means we have to decide correct connection between the contours of the adjacent sections. The branching problem occurs when one or more contours in a section correspond to two or more contours in its adjacent section. The tiling problem means the method of how to decide correct correspondence of points on a set of corresponding contours. In tiling process, a triangular mesh is created by a surface model.

In this paper, the correspondence problem is independently solved by discovering involved relation of contours in adjacent sections. After that we found common tiling rules to construct an arbitrary branching terrain model as tiling triangles from the contour vertices, according to the Constrained Delaunay Triangulation(CDT). In our research, contours are described as **Z**-Map, each point on a contour is a contour vertex, and a contour segment can be constructed by joining two adjacent contour vertices.

Section 2 describes previous approaches. Section 3 outlines the details of our algorithm. Section 4 investigate its an application. Section 5 shows the results with some examples. Section 6 summarizes the work and proposes some future works.

2. Previous Work

In 3D surface reconstruction, the first approach has been introduced by triangulating an area between contours lying on adjacent planes [3]. The solutions of this approach were limited to the case of a single contour on each cross section and there were various restrictions: maximizing the volume of an object [3] and minimizing the surface area [6]. When A):shapes are different between the contours in adjacent sections shown in Fig.1(b-c) and B):the number of contours varies from one plane to another shown in Fig.1(d–e), the methods described in papers [3][6] can not compute the solutions.

Boissonnat et al. [8][9] proposed a solution to the tiling problem based on the Delaunay triangulation of contours. In their papers, the internal Voronoi skeleton (IVS), external Voronoi skeleton (EVS) and the medial axis of contours were extracted to increase the quality of shape representation. Meyers [2] provided multiresolution tiling technique to dispose shapes different between contours in adjacent sections and he used a shaved medial axis (SMA) to divide one-to-many braching into one-to-one cases. Bajaj et al. [1] addressed all of the three problems simultaneously by presenting Optimal Tiling Vertex (OTV) and Edge Voronoi Diagram (EVD) concepts and defining a set of criteria for a desired reconstructed surface. As a new type skeleton method to approximate the EVD, Oliva et al. [10] used angular bi-

Ninth International Conference on Computer Aided Design and Computer Graphics (CAD/CG 2005)
0-7695-2473-7/05 $20.00 © 2005 **IEEE**

sector network(ABN). Their basic idea is to recursively insert intermediate contours for all areas can be triangulated. However, it is a complicated problem to compute IVS, EVS, SMA, EVD and ABN from polygons or curves.

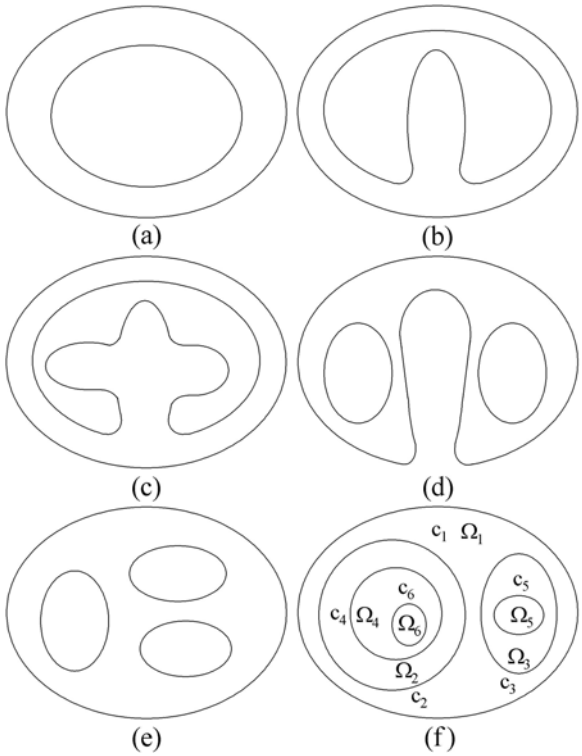

Figure 1. Different shape and topology among contours: (a) simple one-to-one correspondence; (b) one-to-one correspondence with single convolution; (c) one-to-one correspondence with multiple convolutions; (d) double branching with single convolution; (e) triple branching; (f) judgment of correspondence based on sub-domain Ω_i with an exterior boundary contour and some interior boundary contours.

Barequet et al. [4] reduced the problem of branching to a series of piecewise-liner interpolation between each pair of adjacemt slices. In their algorithm, when the remaining clefts are triangulated, the 3D minimum area triangulation technique leads to a defect for convolution shown in Fig.1(b). Sederberg et al. [15] modified the area minimization method for triangulation of branching contours. The algorithm works very well for convolution like Fig.1(b) and simple one-to-two branching. Since a line segment bridge was required to connect two contours, it is difficult for a special one-to-two branching shown in Fig.1(d) and triple or more branching shown in Fig.1(e). In addition, when there are multiple convolutions shown in Fig.1(c), the method is inefficient. This gives us an indication that shape and topology must be divided before reconstruction.

In this paper, we have described how to obtain appropriate surface shape among contours shown in Fig.1(b-e) that can not be solved desirably by using traditional methods.

3. Algorithm

Our terrain reconstruction algorithm consists of three steps: preprocessing, correspondence and tiling. We will describe the details of the three steps in this section.

3.1. Preprocessing

A contour is given as a sequence of gathered or scattered points in a consistent direction — clockwise or counterclockwise. In order to get uniform sample points: First, we approximate the vertices of each contour with a cubic periodic uniform B-spline curve using chord length parameterization [5]. The number of control points (NCP) can be decided by using compact curve fitting algorithm with given accuracy [7]. Then, disperse the cubic periodic B-spline curves into sampling points of uniform density and to generate evenly spaced contour vertices. A good sample is a contour in which the sampling density is (at least) inversely proportional to the distance to other contours [14]. This criterion can be described as follows: when the Euclidean distance D_s from a sample point $\mathbf{P}$ to the nearest sample point is at most s times as far as the distance D_c from $\mathbf{P}$ to the nearest point on other contours. The proportionality constant s is generally smaller than 1. A sample satisfying the criterion is called s-sample.

3.2. Correspondence

Hormann et al. [11] expressed the correspondence method by using nesting tree that is a good guide when any contour does not intersect with other contours. There are only two cases in correspondence relation: In a contour, A):no other contour is involved ($\mathbf{C_5}$, $\mathbf{C_6}$ in Fig.1(f)) or B):one or more contours are involved (Fig.1(a–e)). For a set of $\mathbf{m}$ contours $\mathbf{C_i}$ $(\mathbf{i = 1, \ldots, m})$ (Fig.1(f)), the outermost contour that encloses all the others is defined as *Bottom* $= \mathbf{C_1}$, and the contour with the greatest height is defined as *Top* $= \mathbf{C_m}$. *Bottom* $\mathbf{C_1}$ is the most exterior boundary of enclosed domain $\mathbf{\Omega_0}$, and the other contours divide the domain into $\mathbf{m}$ disjoint sub-domains $\mathbf{\Omega_i}$ $(\mathbf{i = 1, \ldots, m})$. Each sub-domain has one exterior boundary and some interior boundaries and the height of each interior boundary may be different. We denote *NIB* to

note the number of interior boundaries for one sub-domain, then only three cases can be distinguished:

- If *NIB* $= 0$, the sub-domain $\mathbf{\Omega_j}$ has only one exterior boundary $\mathbf{C_j}$. Terrain in here is the *Peak* or the *Bottom of a Basin*. Obviously, *Top* is the *Peak*. It is difficult to distinguish *Peak* from *Bottom of a Basin* because we do not know whether the height of points in the sub-domain $\mathbf{\Omega_j}$ is higher or lower than its exterior boundary $\mathbf{C_j}$. But we can consider the height relationship between $\mathbf{C_j}$ as an interior boundary and its exterior boundary $\mathbf{C_{j-1}}$ of sub-domain $\mathbf{\Omega_{j-1}}$. If $\mathbf{C_j}$ is higher than $\mathbf{C_{j-1}}$, sub-domain $\mathbf{\Omega_j}$ is the *Peak*; otherwise it is the *Bottom of a Basin*.

- If *NIB* $= 1$, the terrain is classified into standard one-to-one correspondence case. Just like *Peak* and *Bottom of a Basin*, if interior boundary $\mathbf{C_i}$ is higher than exterior boundary $\mathbf{C_e}$, sub-domain $\mathbf{\Omega_e}$ is recognized as a *Tableland*. If the height of $\mathbf{C_i}$ is equal to $\mathbf{C_e}$ it is a *Ridge-Peak*; otherwise it is a *Basin*.

- If *NIB* > 1, it's a one to *NIB* branching. Generation of terrain becomes complicated if this case occurs. Because it will be a complex of some *Tablelands*, *Ridge-Peaks*, *Basins* or their mixture. As a special case, one-to-one correspondence can be treated as degenerated one-to-many branching case.

3.3. Tiling

Traditional tiling triangles were always constructed with three vertices: two sequential vertices on a contour and the third vertex on another contour [3] [1] [15]. As mentioned earlier, the condition is a limitation for tiling a domain of one-to-many case and special one-to-one case. Therefore we renounced the rule and supported three tiling vertices that are relatively independent. Each tiling vertex of a tiling triangle may lie on an arbitrary contour. Our tiling algorithm is enforced in each sub-domain $\mathbf{\Omega_i}$ $(\mathbf{i} = \mathbf{1}, \ldots, \mathbf{m})$ and it is illustrated in detail as follows:

Searching the start tiling edge In a sub-domain $\mathbf{\Omega}$, the closest vertex pair between exterior boundary contour vertices and interior boundary contour vertices are detected. In our method, when the two closest vertices are decided, the edge connecting them is set as the start tiling edge. Obviously, the start tiling edge must lie in the sub-domain $\mathbf{\Omega}$ and it does not intersect with any contour segment except for their end points. Then we define the start tiling edge e_0 (Fig.2(a)) as the vector $\overrightarrow{P_sP_e}$. P_s lying on an exterior boundary contour is the start point and P_e lying on an interior boundary contour is the end point.

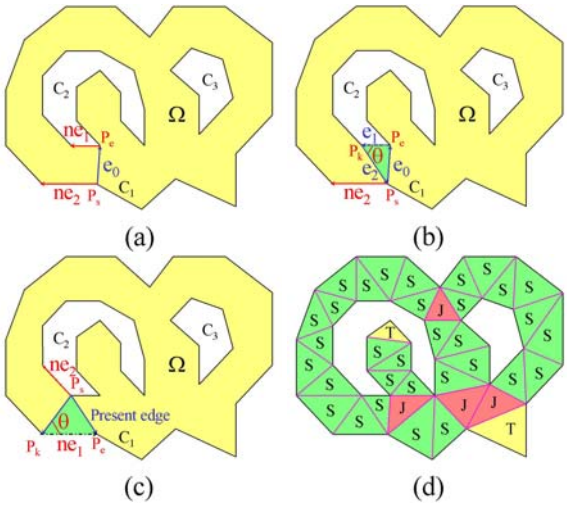

Figure 2. Tiling triangles generation: (a) searching the start edge; (b) constructing the first tiling triangle; (c) promoting of triangulation; (d) CDT represented with terminal(T), sleeve(S) and junction(J) triangles.

If *NIB* is equal to 0, one of the segments of an exterior boundary contour may be selected as the start tiling edge. We assume that all contours are oriented clockwise, and each contour segment and one of the vertices lying on its right hand side will construct a triangle. To select an appropriate vertex, the angle between contour segment and a vertex is calculated. Comparing the maximized opposite angles among all contour segments, the segment with the maximum opposite angle will be set as the start tiling edge. As a vector, the direction of the start tiling edge is opposite to the direction of contour.

Constructing the first tiling triangle There are two contour segments ne_1 and ne_2 lying on the left hand side of e_0 are shown in Fig.2(a). We call the two contour segments limited edges. The contour segment ne_1 has a overlapping vertex to P_e and ne_2 has a overlapping vertex to P_s. A contour vertex P_k that lying on the left hand side of e_0, lying on the ne_1 or its left side more, and lying on ne_2 or its right side more is searched to construct the first tiling triangle, by maximizing the opposite angle θ of the start tiling edge e_0 (Fig.2(b)). The vector $\overrightarrow{P_eP_k}$ is the second tiling edge e_1 (In Fig.2(b), It overlapping on ne_1) and vector $\overrightarrow{P_kP_s}$ is the third tiling edge e_2. Each edge of the first tiling triangle will not intersect with any contour segment except for their end points. In other words,

the area of first tiling triangle mush be portion of sub-domain $\mathbf{\Omega}$ because of the following reasons: a): the start tiling edge is the shortest distance between the exterior boundary contour C_1 and one of the interior boundary contours C_2. b): the maximum angle θ at P_k guaranteeing the first tiling triangle is a Constrained Delaunay Triangulation(CDT) [12], which means that, there are not sample points in the circumscribed circle of the triangle. The proof can be found in [13].

Promoting of triangulation Start promotion of triangulation from the last tiling edge e_2 of the first tiling triangle and its two limited edges. The tiling edge e_2 is called a *present* edge. If the edge overlaps on a contour segment or any other tiling edge, the *present* edge will be exchanged for the next tiling edge and e_2 will be eliminated. Otherwise Constrained Delaunay Triangulation is performed by using maximization rule of opposite angle. Here, vertex P_k must lying on the right hand side of *present* edge, lying on ne_1 or its right hand side and lying on ne_2 or its left hand side (Fig.2(c)). Next, connect two terminal points of the *present* edge to a new vertex, so that two new tiling edges can be generated. The direction of the new tiling edges are defined as vectors $\overrightarrow{P_sP_k}$ and $\overrightarrow{P_kP_e}$. In Fig.2(c) $\overrightarrow{P_kP_e}$ overlapping on ne_1. Then, substitute original *present* edge for the last tiling edge. Repeat this step until any new *present* edge is not found—the sub-domain $\mathbf{\Omega}$ is divided into some Constrained Delaunay Triangulation shown in Fig.2(d).

In CDT, each vertex is called a "tiling vertex" and each edge is called a "tiling edge". Three tiling vertices may lie on one, two or three contours, and a tiling edge and a contour segment may overlap. According to overlapping circumstance of tiling edges and contour segments, tiling triangles are classified into three groups shown in Fig.2(d): *Terminal triangles*: only one tiling edge don't overlapping on contour segment. *Sleeve triangles*: only one tiling edge overlapping on contour segment. *Junction triangles*: no tiling edge overlaps on any contour segment.

In our method, medial axis is unused as support information. In other words, our method can directly obtain tiling triangles. Therefore, it is briefer and more efficient than traditionary medial axis method and minimizing surface method. When a sub-domain $\mathbf{\Omega}$ is composed of one exterior boundary contour with N_e vertices and some interior boundary contours with N_i vertices, the worst time computational complexity is $O(N_eN_i + (N_e + N_i)^2)$. Because we used dynamic limited edges that can be seen as linear change, actual time complexity has closely resembled to $O(N_eN_i + k(N_e + N_i))$. Suppose $N = N_e = N_i$, we can briefly describe it to $O(N^2 + 2kN)$. k ($k < N$) is the linear factor.

4. Application

When a sub-domain $\mathbf{\Omega}$ with m boundary contours are divided by junction triangles, its curve mesh model can be generated. For three vertices of a junction triangle, only three cases are examined: 1) All vertices lie on different contours; 2) Two vertices lie on the same contour; 3) All vertices lie on the same contour. Except for 1), there must be *Canyon* or *Ridge* in 2) and 3). If all the triangles of 2) and 3) are used, *multiple Canyons* and *Ridges* will be extracted and the curve mesh model will become too detailed with some troublesomely small wrinkle. Therefore, some junction triangles are changed into sleeve triangles, and it is called elimination operation. Our elimination rule is shown in Fig.3: (a) corresponds to the case 2) and (b) corresponds to the case 3).

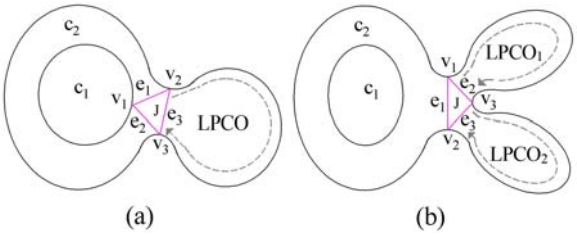

Figure 3. Elimination rules establishment by presenting one or more *comparison* parameters r_i and a *resolution* parameter w that can be specified by user.

In Fig.3(a), tiling edges e_1 and e_2 have a common vertex v_1 lying on contour C_1. Other two vertices v_2 and v_3 lying on contour C_2 construct tiling edge e_3, which divides the contour C_2 into two portions: one portion lying on the same side of e_3, and the other lying on the opposite side. First, we define the length of the partial contour lying on the opposite side as $LPCO$ and the lengths of three tiling edges are L_1, L_2 and L_3, respectively. Then we define $r = \frac{LPCO}{(L_1+L_2)}$ as the *comparison* parameter and $w(w \geq Max[\frac{L_3}{(L_1+L_2)}, \frac{L_2}{(L_1+L_3)}, \frac{L_1}{(L_2+L_3)}])$ as the *resolution* parameter that can be specified by a user. If $r < w$, this junction triangle would be eliminated; otherwise it would be reserved. Fig.3(b) is similar to Fig.3(a) while the difference is that there are two *comparison* parameters $r_1 = \frac{LPCO_1}{(L_1+L_3)}$ and $r_2 = \frac{LPCO_2}{(L_1+L_2)}$. If $r_1 < w$ or $r_2 < w$, the junction triangle would be eliminated; otherwise it would be reserved. The elimination operation shown how to eliminate troublesomely small wrinkle by user specified parameters.

5. Results

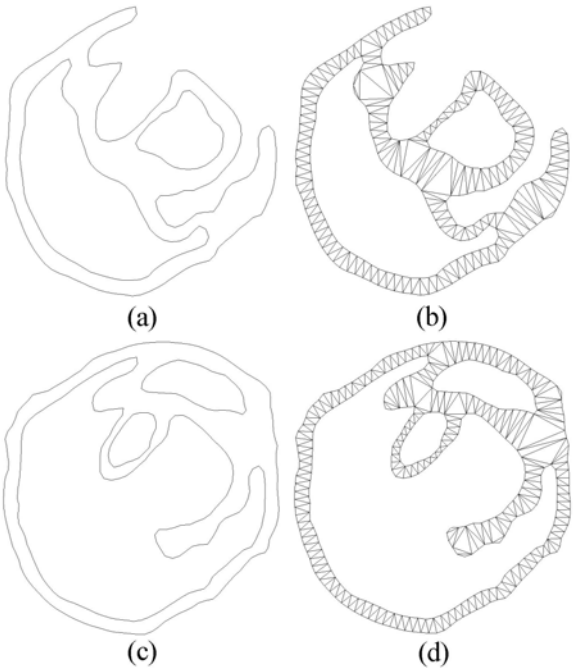

Figure 4. Tiling result of typical contours

We applied our technique to Mt. Iwate of Japan. The results are shown in Fig.4 to Fig.8. In our research, the *resolution* parameter w was defined as 1.

Fig.4(a) and (c) shown two typical layering contours, (b) and (d) are their tiling results that used our method.

In Fig.5(a), the domains I are *Ridges* and II is *Canyon*, shown in (b). In Fig.5(c), the domains III are *Canyons*, IV is a *Basin* and V are *Ridge-Peaks*, shown in (d). *Canyon line* or *Ridge line* can be approximated by connecting the center q of sleeve triangle edges between a junction triangle and a terminal triangle (Fig.6(a) – a brief magnifier inside the quadrangle shown in Fig.5(a)). The height of q were approximated by linear interpolation as follows:

$$h_q = \frac{h_l d_{qu} + d_{ql} h_u}{d_{qu} + d_{ql}} \tag{1}$$

where, d_{qu} is the shortest distance from q to its upper contour C_1 with height h_u, and d_{ql} is the shortest distance from q to its lower contour C_2 with height h_l (Fig.6(b)).

Fig.7(a) shows the contours model that were approximated by cubic periodic uniform B-spline curves with chord length parameterization [16]. Fig.7(b) is the curve mesh model with reserved junction tiling triangles, *Canyon lines* and *Ridge lines*.

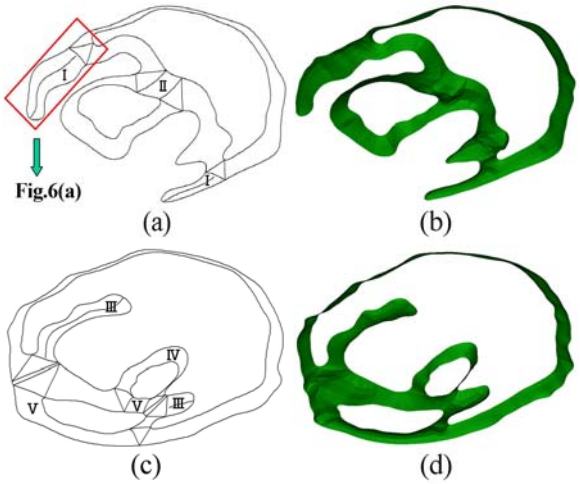

Figure 5. Complicated curve mesh models and their corresponding terrain: (a-b) double branching case with *Ridges* and a *Canyon*; (c-d) triple branching complex with *Canyons*, a *Basin* and *Ridge-Peaks*.

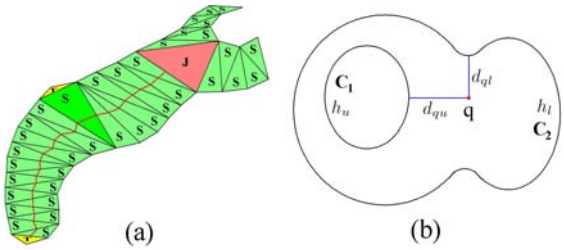

Figure 6. Obtain information of *Ridge line*

Fig.8 shows two terrain surface models that were expressed as polyhedron shell based on tiling result of Fig.7(a) and Fig.7(b), respectively. Obviously, we can discriminate *Peaks*, *Bottom of a Basin*, *Tablelands*, *Ridge-Peaks*, *Basins*, *Canyons* and *Ridges* in Fig.5 and Fig.8.

6. Summary and future work

We have presented a new algorithm to reconstruct a terrain from contours. First, we processed primitive data by extending the concept of s-sample with universality. Then, a polyhedron model with triangular meshes was generated by constructing CDT tiling triangles. In order to obtain quality curve mesh model, we proposed the *comparison* parameter and *resolution* parameter method. At last, We demonstrated our algorithm by using practical contours.

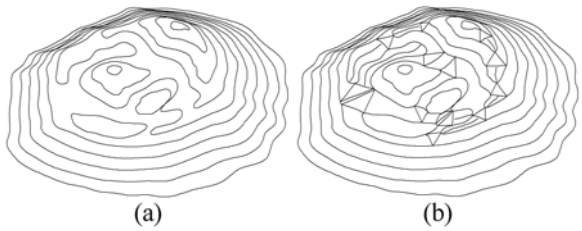

Figure 7. Contours model of Mt.Iwate of Japan and its curve mesh model.

As our future work, it will be important and prerequisite for precision data generation to fit parametric surfaces to curve mesh. If no tiling edge exists in a curve mesh (Fig.1(a)), the curve mesh is a closed area with two boundaries that can be expressed by a periodic B-spline surface. Otherwise, only three types curve mesh need to be expressed by parametric surfaces: 2-boundary (Fig.5(a).I and Fig.5(c).III), 3-boundary (junction triangles) and 4-boundary (Fig.5(a).II and Fig.5(c).IV and V). We will pay attention to the continuity between adjacent parametric surfaces and estimate the accuracy of generated surfaces.

Acknowledgement
This work was supported partly by the Science and Technology Research Program of Iwate Prefectural Government.

References

[1] Chandrajit L.Bajaj, Edward J.Coyle and Kwum-Nan Lin: Arbitrary Topology Shape Reconstruction from Planar Cross Sections. Graphical Models and Image Processing, Vol.58, No.6, pp.524-543, 1996.

[2] David Meyers: Reconstruction of Surface From Planar Contours. PhD thesis, University of Washington, 1994

[3] E.Keppel: Approximating Complex Surfaces by Triangulation of Contour Lines. IBM J. Res. Develop., 19:2-11, January 1975

[4] Gill Barequet and Micha Sharir: Piecewise-Linear Interpolation between Polygonal Slices. Computer Vision Image Understanding, 63:251-272, 1996

[5] G.Farin: Curves and Surfaces for Computer Aided Geometric Design-A Practical Guide. Second Edition, Academic Press, Inc. 1990

[6] H.Fuchs, Z.M.Kedem and S.P.Uselton: Optimal surface reconstruction from planar contours. Communications of the ACM, 20(10):693-702, Oct., 1977

[7] Hyungjun Park and Kwangsoo Kim: Smooth surface approximation to serial cross-sections. CAD, Vol.28, No.12, pp.995-1005, 1996

[8] Jean-Daniel Boissonnat: Shape Reconstruction from Planar Cross-Sections. Research report of INRIA, No.546, June, 1986

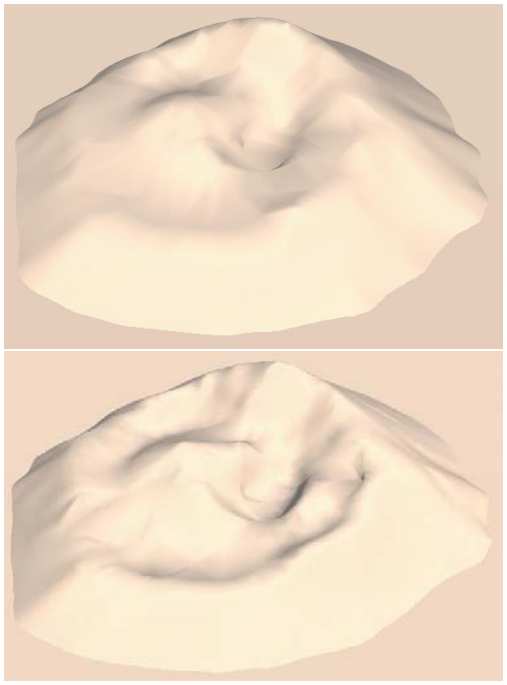

Figure 8. Terrain models of Mt.Iwate of Japan.

[9] Jean-Daniel Boissonnat and Bernhard Geiger: Three Dimensional Reconstruction of Complex Shapes based on The Delaunay Triangulation Planar. Research report of INRIA, No.1697, April, 1992

[10] J-M.Oliva, M.Perrin and S.Coquillart: 3D Reconstruction of Complex polyhedral Shapes from Contours using a Simplified Generalized VoronoiDiagram. Eurographics'96, Vol.15, No.3, pp.C397-C408, 1996.

[11] Kai Hormann, Salvatore Spinello and Peter Schröder: C^1-continuous Terrain Reconstruction from Sparse Contours. VMV 2003, Munich, Germany, November 19-21, 2003.

[12] Marsahll Bern and David Eppstein: POLYNOMIAL-SIZE NONOBTUSE TRIANGULATION OF POLYGONS. International Journal of Computational Geometry & Applications, Vol. 2, No. 3, pp.241-255, 1992.

[13] Mark de Berg, Marc van Kreveld, Mark Overmars and Otfried Schwarzkopf: Computational Geometry: Algorithms and Applications. published by Springer-Verlag, 2000.

[14] Nina Amenta, Marsahll Bern and Manolis Kamvysselis: A New Voronoi-Based Surface Reconstruction Algorithm. Siggraph'98, pp.415-421, 1998.

[15] Thomas W.Sederberg, Krzysztof S.Klimaszewski, Mu Hong and Kazufumi Kaneda: Triangulation of Branching Contours using Area Minimization. International Journal of Computational Geometry and Applications, Vol.8, No.4, pp.389-406, 1998.

[16] Zhiyi Zhang, Kouichi Konno and Yoshimasa Tokuyama: 3D Model Generation of Mountain Terrain Based on Periodic B-Spline Curves. NICOGRAPH International 2004.