Contour Interpolation and Surface Reconstruction of Smooth Terrain Models

Jianyun Chai* Takaharu Miyoshi† Eihachiro Nakamae‡

Sanei Co. Hiroshima Institute of Technology

Abstract

Interpolating contours and reconstructing a rational surface from a contour map are two of essential problems in terrain modeling. They are often met in the field of computer graphics and CAD systems based on geographic information systems. Although many approaches have been developed for these two problems, one difficulty is still remained open. That is how to ensure the reconstructed surface both to be smooth globally and to coincide with the given contours exactly simultaneously.

In this paper we solve the two problems in a unified framework. We use gradient controlled partial differential equation(PDE) surfaces to express terrain surfaces, in which the surface shapes can be globally determined by the contours, their locations, height and gradient values. The surface generated by this method is accurate in the sense of exactly coinciding with the original contours and smooth with C^1 continuity everywhere. The method can reveal smooth saddle shapes caused by surface branching of one to more and can make rational interpolated sub-contours between two or more neighboring contours.

CR Categories and Subject Descriptors: I.3.5 [Computer Graphics]: Computational Geometry and Object Modeling - surfaces and object representations; G.1.6[Numerical Analysis]: FEM

Additional Keywords: PDE surfaces, Terrain modeling, Shape reconstruction, Contour interpolation

1 INTRODUCTION

Interpolating contours and reconstructing a rational surface from a contour map are two of essential problems in terrain modeling. A Terrain surface is also called a height field that can be described mathematically by a bivariate elevation function. Since their simplicity, contour maps have employed as one of major forms to record terrain information for a long time. As the progresses of computer graphics(CG) and geographic information systems(GIS), polygonal meshes such as digital terrain models(DTM) and triangulated irregular networks(TIN) have become fundamental techniques in many fields. Nowadays, CAD systems based on GIS are required to deal with both of contour maps and polygonal surfaces, and convert them to each other. Although it is easy to generate contours from a polygonal surface, generally, reconstructing a rational surface from the contours is not easy. Because the contours in a map are only the section data at several height levels, some important surface features such as saddles, tops and bottoms may be skipped. Special care should also be taken to the places where the shapes of neighboring contours look very different. These key terrain shapes have to be revealed by contour interpolation or extrapolation when reconstructing the surface. Here, the contour interpolation means to find the sub-contours or the height of points in the region bounded by the neighboring contours of different levels; while the contour extrapolation is for those outside the region. For example, to form a smooth saddle caused by surface branching are a typical interpolation problem; while to build tops and bottoms are classified into extrapolation problems.

In this paper we concentrate our discussion only on the contour interpolation and its associated surface reconstruction problems. We propose a new method to reconstruct smooth surfaces from contours. It ensures the surfaces to be C^1 continuos even in the places across the contours. On these surfaces, high-quality interpolated sub-contours can be obtained.

The remainder of this paper is organized as follows. In the following two sections, we briefly review related work and describe the main idea of this paper. Sections 4 and 5 give the fundamental theory and the implementation of our method. In section 6 we show several examples. Conclusions are listed in the last section.

2 RELATED WORK

The problem of reconstructing a surface from a collection of contours has been intensively studied in the recent two decades [16]. In the case of terrain models, it consists of three major subjects. The first subject is to set up the correspondence of the vertices or segments on the contours of defferent height levels, so that these contours can be triangulated to a polygonal surface. Graph search[12, 10] and curve mapping[7] are two typical methods for this subject. To get a "good" surface, some metric cost functions such as "minimum area" [10] , "minimum span length"[7] and "total sum of edge length"[20] were used. The second subject of the problem is to deal with surface branching. On a terrain surface, only the branching from one contour to many contours should be considered. A simple way to do it is to introduce some bridges to link the contours with same height, thus these contours can be combined into a equivalent contour. Then the methods for tow contours can be used [9, 18]. There are still many other methods for "many to many" branching problem[6, 1, 2], but they are beyond the scope of this paper. The third subject of the problem is to fit the polygonal surface, obtained in the procedures for the first two subjects, by a smooth surface. Bezier patches[15], curved network[17] or other

*3-13-26-402 Kagamiyama Higashihiroshima, 739-0046 Japan, Email:chai@sanei.co.jp

†2-1-1 Miyake, Saeki-ku, Hiroshima 731-5193 Japan, Email:miyoshi@cc.it-hiroshima.ac.jp

‡2-1-1 Miyake, Saeki-ku, Hiroshima 731-5193 Japan, Email:nakamae@cc.it-hiroshima.ac.jp

type curved patches in CAGD[19] can be produced from the polygonal surface.

Most of the methods mentioned above confine themselves to using

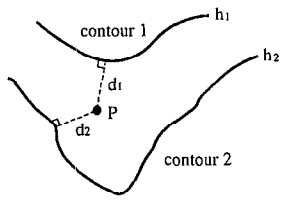


Figure 1: Contour interpolation based on the ratio of distances.

only the vertices of the given contours. However, it is possible to build a better surface if interpolated extra vertices or sub-contours can be employed [7, 11]. An automatic method to interpolate the height of any vertex located between two contours is given in[11], which estimates the height by the ratio of distances from the vertex to the contours(see Fig.1). A similar interpolation scheme is used in [14], but it is implemented on a scanned raster map. Morphological transforms in the image space[3] is also used for contour interpolation. Using partial differential equations(PDE) for surface modeling were explored in[5]. An interesting application of the PDE surface for terrains can be found in[13]. A common problem of these interpolation schemes is that they do not ensure the reconstructed surface to be smooth, especially in the places across the given contours. We present a new solution to the contour interpolation and its associated surface reconstruction. It is basically different from the methods as described in the following section.

3 PROPOSED METHOD

We assume that a terrain surface bounded by the contours of two successive height levels is smooth so that its height function can be governed by partial differential equations. By solving these PDEs with their boundary height and gradient conditions, we obtain a piecewise linear polygonal terrain surface. Then sub-contours can be drawn by sampling the resultant surface.

An apparent merit of the PDE method over the methods such as CAGD and curved networks[19, 17] is that the surface smoothness can be ensured globally. Therefore, matching the segments or vertices between the contours of different height level is unnecessary in our method. The method can also automatically deal with surface branching problems met in terrains. Unlike the previous methods[11, 3, 14, 13], our method takes the gradient conditions across contours into consideration of the PDE surface. Thus two neighboring surfaces sharing a common boundary contour can be smoothly connected. In other words, the effects of the contours outside the interpolating region can be reflected in the contour interpolation.

4 PDE TERRAIN SURFACE

4.1 PDE Surface

A PDE surface is a bounded surface expressed by partial differential equations, that is, it is a kind of implicit surfaces. In most cases, PDE surfaces usually have no analytical solution. Thus some numerical method has to be employed. The general equation form for a PDE surface defined on a region D in two dimensional Euclidean space (u, v) can be written as,

$$P(u,v)h(u,v) = F(u,v) \tag{1}$$

$$\text{constrained by } (h, \frac{\partial h}{\partial n}) \mid_{Bl},$$

where $P(u, v)$ is an operator including partial differential operation; $h(u, v)$ is the surface height at location (u, v), and $F(u, v)$ is the function of forcing sources. The constraint term $(h, \partial h/\partial n) \mid_{Bl}$ presents the conditions of height and normal gradient on boundary lines Bl, and n is the normal vector of Bl on uv-plane. It should be noted that the shapes of the PDE surface depend upon not only the boundary conditions but also the control terms of $P(u, v)$ and $F(u, v)$. A PDE surface is called a controlled PDE surface in the cases operator $P(u, v)$ varies depending on the coordinates (u, v) and/or source term $F(u, v)$ is non-zero.

According to the classical PDE theory, the first kind of boundary conditions(height values) and the second kind of boundary conditions(derivative values) can not be imposed at the same place simultaneously, otherwise the equations will be over-constrained. The boundary constraints should be of form $(h \text{ or } \partial h/\partial n) \mid_{Bl}$. In order to smoothly connect two surfaces sharing a common boundary curve, the surfaces must have the equi-height and cross derivative values along that boundary curve. Therefore, both the height and derivative boundary conditions have to be imposed on the surface. Conventional PDE surfaces do not accept such boundary constraints. Bloor et. al. [5]showed some types of PDE surfaces and discussed the effects of the control terms on the shapes of the PDE surfaces, but they did not discuss how to control the surface shapes by boundary gradient conditions. In this paper, we explore a new type PDE surface which satisfies both kinds of boundary conditions.

4.2 Gradient Controlled PDE Surface

4.2.1 Analogy between height field and electric field

Observing a 2D electric field generated by some electrodes(see Fig.2), one can find that the potential contours in the field have similar properties of terrain contours. The electric field distributed

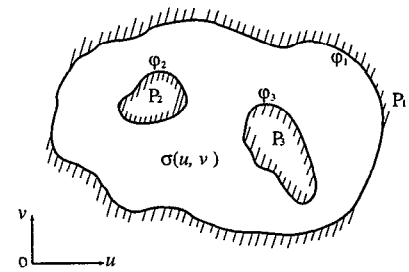


Figure 2: A 2D electric field in conductive region. P_1, P_2, and P_3 are electrodes, $\sigma(u, v)$ conductivity, and φ_1, φ_2, and φ_3, boundary potentials.

in a conductive region obeys the following,

$$div\,(\sigma(u,\,v)grad\varphi(u,\,v)) = 0 \tag{2}$$

$$\text{constrained by } (\varphi \text{ or } \frac{\partial \varphi}{\partial n}) \mid_{Bl},$$

where $\varphi(u, v)$ is a scalar potential function, $\sigma(u, v)$ is material conductivity, div is a 2D divergence operator, and $grad$ is a 2D gradient operator. It is well known that electric flux lines can be used to display the field distribution. The flux lines satisfy the following equation,

$$\frac{du}{E_u} = \frac{dv}{E_v}, \tag{3}$$

where E_u and E_v are two components of the intensitv of electric field E = $grad\varphi(u, v)$, and du and dv are differentials of u and v, respectively. The flux lines and the contours are two set of curves which are orthogonal each other everywhere. Some properties of the electric field governed by Eq.2 can be expressed by the following:

(a) Non-intersection: any two of contours or any two flux lines never intersect each other.

(b) Monotony: the potential varies monotonically along with any flux line.

Now let us turn our subject to a height field obeying the same equation as Eq.2. We use the terms of gradient of height G = $grad\,h(u, v)$, slope lines, and the local surface parameter in the height field. The slope lines describe the slope directions of the surface, which obey the same equation of the flux lines. It should be noted that three assumptions are used here for the height field.

(1) Terrain contours do not intersect each other; no consideration of cliffs;

(2) Along a slope line between two neighboring height levels, the height varies monotonously;

(3) Slope lines do not meet nor branch out each other.

Therefore, no local peak nor bottom exists between every two neighboring contours.

The analogy of the height field and the electric field above mentioned offers us a new standpoint to measure and control terrain surfaces.

4.2.2 Relating local parameter with boundary conditions

The following problems should be solved when a PDE surface governed by Eq.2 is applied to a terrain model: (1) how to determine the local parameter $\sigma(u, v)$ of the surface and (2) how to force the surface to satisfy both height and gradient conditions on its boundary contours. Fig.3 shows a PDE surface with a uniform local parameter between two contours. The surface is smooth, and its shape

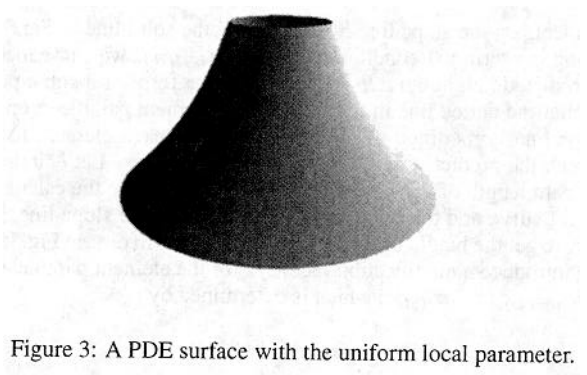


Figure 3: A PDE surface with the uniform local parameter.

reflects the inherent curved pattern of the Laplace type PDE surface. The shape can be changed by adjusting the distribution of the surface local parameter $\sigma(u, v)$. An extended form of Eq.2 including both height and gradient boundary conditions on the contours is expressed as:

$$div\,(\sigma(u, v)gradh(u, v)) = 0 \qquad (4)$$

$$\text{constrained by } (h \text{ and } \frac{\partial h}{\partial n})\,|_{Bl}\,,$$

where $(h\,and\,\partial h/\partial n)\,|_{Bl}$ means both h and $\partial h/\partial n$ are imposed simultaneously. To avoid the equations to be over-constrained, we use the boundary gradient conditions to adjust the surface local parameter. It is clear that the surface of Eq.4 is not a conventional PDE surface. In fact, we have to solve an inverse PDE problem to determine the surface satisfying Eq.4. As far as we know, there is no analytical solution to this problem. Here we solve this problem in two steps. The main idea is to separate Eq.4 to a conventional problem as Eq.2 and the additional constraints of boundary gradients on the contours. In the first step, the former is solved by employing the conventional method, and in the second step the latter is used to modify the local parameter $\sigma(u, v)$ on the surface obtained in the first step. The modified local parameter is fed back into the former to renew the surface. These two steps are performed iteratively until the surface shape converges.

The key point of this method is how to relate the local parameter $\sigma(u, v)$ with the boundary conditions $(h\,and\,\partial h/\partial n)\,|_{Bl}$. It requires each point on the surface to be explicitly linked to the surface boundaries. We build such relations by using slope lines on the surface. Given a point on the surface, the slope line passing through this point can be traced toward the boundary contours. Thus the height of the surface on the slope line can be found. We predict the height along the slope line from its terminal height and gradient values. Comparing the calculated and the predicted height distributions on the slope line, the values of the local parameter $\sigma(u, v)$ along the slope line is modified. The modification can be properly done over the whole surface by generating a suitable number of slope lines as described in the next section.

5 IMPLEMENTATION

We first explain the input and output data used in our algorithm, and then describe the steps to solve the PDE. As mentioned in the last section, we solve the PDE in two steps. For the first step, we discuss on the variational problem of the PDE surface and its discretization, for the second step, the algorithm to modify the surface local parameter $\sigma(u, v)$. Concerning the connection of terrain consisting of several contours, we introduce a simple method to estimate the gradients on these contours.

5.1 Definition of Input and Output

A set of contours consisting of polylines are used as the input data. In addition to their locations and height, the gradients across them can be given explicitly or estimated in the process refining the surface shapes. Such conditions ensure that the reconstructed terrain surface is C^1 continuous across the contours, even though the surface is formed one by one in every sub-region bounded by neighboring contours. We consider two cases of boundary conditions

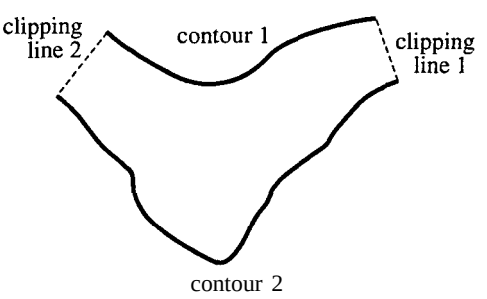


Figure 4: Boundary conditions.

(see Fig. 4). On the boundary parts of contours, height h and/or gradient $\partial h/\partial n$ are given. On the clipping lines, normal gradient condition is set to zero. That means the clipping lines should be slope lines. Because of the monotonic property of the surface, the boundary conditions should be carefully imposed. The slope angle at any point on the boundary is limited to the range of [0, n/2). If the gradient values on the contours are not given, default gradient

conditions described later are used. The output data consists of a number of triangular segments. These surfaces can be expressed as discretized C^1 smooth terrain surface, from which the sub-contours and the height of any point on the surface are sampled.

5.2 Variational Problem for FEM and its Discretization

We solve a sub-problem of Eq.4 in the first step, in which only the direct boundary conditions are concerned. A variational problem corresponding to the sub-problem is to find a function h that minimizes the energy function W in the following equation,

$$W(h) = \int_D \frac{\sigma}{2}[(\frac{\partial h}{\partial u})^2 + (\frac{\partial h}{\partial v})^2]ds \qquad (5)$$

constrained by boundary height conditions,

where ds denotes the differential of area in region D. In order to apply FEM to the variational problem, D is discretized into a triangular mesh. For contour maps, that can be done simply. We first triangulate the contours on the horizontal plane to get a rough mesh, then subdivide it into a properly dense mesh; e.g., the elements in the mesh can be split repeatedly until every edge in the mesh becomes shorter than a given length. It is assumed that the surface height varies linearly and the local parameter is kept at a constant value in every triangular element. The details about the FEM solution can be found in [21, 4].

We use the first order triangular elements e_i in the FEM, each element has an individual constant vector of its height gradient. To improve the estimation of the surface gradient, we define the gradient on a vertex V , $\mathbf{G}_V$, by the averaged vector of the gradients in all element containing the vertex V. That is

$$\mathbf{G}_V = \frac{\sum \alpha_i \mathbf{G}_i}{\sum \alpha_i}, \qquad (6)$$

where $\mathbf{G}_i$ is the gradient vector in element e_i, and α_i is the interior angle at V of e_i. These gradients are used to rearrange the gradient in every element to a linear distribution.

5.3 Modification of Local Parameter

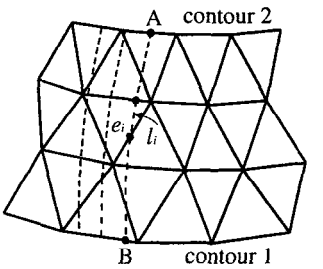
Because of the discretization of the surface, the distributed local parameter on the surface is approximated by a constant in each element. The initial value of every element parameter is set to unit value. Each parameter given in the first step is modified by the following three steps.

(1) Searching slope lines on the triangular mesh

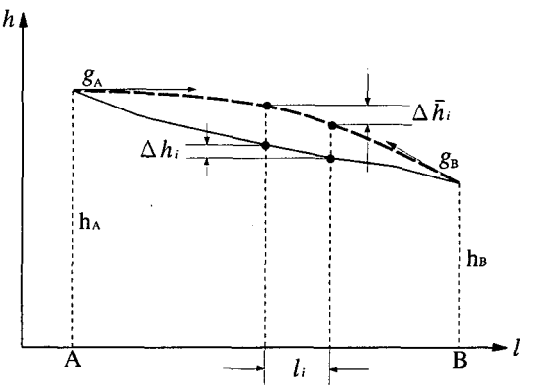
The parameter equations for flux lines (see Eq.3) can be used for slope lines by replacing the terms E_u and E_v to G_u and G_v, respectively. A slope line in each element can be approximated by a linear segment oriented in the direction of the gradient at its entrance or starting point. It leads to a simple algorithm. We found it is enough for our purpose although a more accurate method can be found in [8]. For each element we generate a slope line passing through its center, thus every element has at least one slope line passing through it.

(2) Modifying element parameters on a slope line

Suppose a slope line starts from point A and ends to point B, both A and B are on the boundary contours (see Fig.5(a)). Let h_A, h_B, and g_A, g_B denote the height and normal gradient values on points A and B. If point A (or B) falls on a contour with the default boundary gradient conditions, i.e. if the gradient values are not given, the gradient at the point $g_A(or g_B)$ is set to the averaged gradient along the slope line. The slope line passes through a number of triangular elements indicated by $e_i(i = 1, 2, ..., n)$. The height of the



(a)



(b)

Figure 5: The surface height along a slope line. (a) a slope line on the calculated surface; (b) the surface height along the slope line. solid line: calculated height curve; broken line: predicted height curve.

elements on the slope line h is shown by the solid line in Fig.5(b). Using the terminal conditions (h_A, g_A, h_B, g_B), we can generate a predicted height curve $\bar{h}$ expressed by the form of a cubic polynomial(the dotted line in Fig.5(b)). Each element parameter on the slope line is modified to force the height of each element to approach the predicted height distribution as follows. Let l_i indicate the path length of the slope line in element e_i. From the calculated height curve and the predicted height curve on the slope line, it is easy to get the height differences Δh_i and $\Delta \bar{h}_i$ in e_i (see Fig.5(b)). We introduce a modification factor fi for the element parameter in e_i, $\sigma_{i(new)} = f_i \sigma_{i(old)}$, which is determined by

$$f_i = \frac{\Delta h_i}{\Delta \bar{h}_i}. \qquad (7)$$

(3) Modifying element parameter in an element

Because an element may have plural slope lines passing through it, it is reasonable to combine the contributions from these lines into one factor. Suppose that l_{ij} is the path length and f_{ij} is the modification factor of the jth slope line in the element e_i. The combined modification factor f_i of the slope lines in the element is obtained by

$$f_i = \frac{\sum l_{ij} f_{ij}}{\sum l_{ij}}, \qquad (8)$$

and the element parameter is modified by

$$\sigma_{element(new)} = f \sigma_{element(old)}. \qquad (9)$$

After all element parameters are renewed, they are returned to the first step to get a new surface and new element parameters. These

two steps are iterated until the calculated surface converges within a permissible error ϵ,

$$\frac{\sum \mid \triangle g_e \mid}{\sum \mid g_e \mid} \leq \epsilon, \tag{10}$$

where g_e is the norm of the gradient vector in element e, and $\triangle g_e$ is the difference of g_e between two successive iterations.

5.4 Integration of surfaces

Since a contour map often consists of a large number of contours, to reduce the computation cost, it is better to separately calculate every surface bounded by the neighboring contours of two height levels. Our method ensures that the surface integrated from these sub-surfaces is C^1 continuous, if proper gradients on the contours can be given. In practice, however, contour maps do not provide such gradient information directly. We estimate the gradients on the contours as follows.
(1) Set the default gradient conditions on every contour, and calculate triangular elememt parameters on each surface severally.
(2) combine the surfaces into one surface of C^0 continuity, and calculate the gradients on the contour vertices by using Eq.6.
The estimated gradients on the contours are then used to constrain the surfaces, so that they can be integrated into a C^1 continous terrain surface.

6 EXAMPLES

We give six examples to demonstrate the features and capabilities of the gradient controlled PDE surface. The interpolated sub-contours and slope lines are shown along with the reconstructed surfaces. Since the FEM is employed in our implementation, the surfaces are depicted by triangular meshes. We intentionally use flat shading for all perspective views to examine the surfaces in detail. The interpolated sub-contours and slope lines are also displayed by polylines. Fig. 6 shows the difference between a free-controlled PDE surface (Fig. 6(a)) and a gradient controlled PDE surface (Fig. 6(b)) with the same contours. The uniform local parameter is used for the surface in Fig. 6(a), and the default gradient boundary conditions(see section 5.3) are used for the surface in Fig. 6(b). We can see that the control of surface gradients prevents the surface from unexpected bending. Fig. 7 shows how the shape of a gradient controlled PDE surface can be globally controlled by its gradient boundary conditions. The surfaces in Figs. 7(a) and (b) have the same height but different gradient values on their boundary contours.
To confirm the capabilities of the proposed method, we display two kinds of complicated models. Fig. 8 illustrates a surface with two branches, in which the default gradient conditions are set on the boundary contours (thick lines). The interpolated sub-contours and the slope lines are given in Fig. 8(a) and the reconstructed surface is displayed in Fig. 8(b). Fig. 9 shows a more complicated surface consisting of three branches. The saddle shapes due to the surface branching are recovered automatically. Fig. 10 shows a surface generated from two contours with very different shapes. In this case, default gradient conditions are set to the boundary contours. The curved ridge lines on the surface can be clearly observed from the gathering of the slope lines. In Fig. 11, we show two surfaces reconstructed from a part of a real map. In Fig. 11(a) only C^0 continuity is satisfied, while in Fig. 11(b) C^1 continuity is ensured. The interpolated sub-contours on the surface in Fig. 11(b) are shown in Fig. 11(c). From this example, we can see that the proposed method is suitable to produce high quality sub-contours and smooth terrain surfaces from contour maps.

7 CONCLUSIONS AND FUTURE WORK

We have described a new method to interpolate sub-contours and reconstruct terrain surface from a contour map. The advantage of this method is that the surfaces generated are more accurate and smooth.
The main contribution of this paper is that a new type of PDE surface, the gradient controlled PDE surface, is set up to express rational terrain surfaces. Differing from conventional PDE surfaces, it can satisfy both the height and gradient boundary conditions.
The effectiveness and usefulness of the proposed method have been confirmed in several examples. They show the methid is suitable to capture complex terrain shapes.
In future work we expect to improve the estimation of the gradients on contours. Another interesting issue to be explored is to extend the gradient controlled PDE surface by using non-zero forcing sources. Such a PDE surface can be expected to handle the extrapolation of terrain models.

References

[1] C.L. Bajaj, E.J. Coyle, and K.N. Lin. Arbitrary topology shape reconstruction from planar cross sections. *Graphical models and image processing*, 58(6):524–543, 1996.

[2] G. Barequet and M. Sharir. Piecewise-linear interpolation between polygonal slices. *Computer vision and image understanding*, 63(2):251–272, 1996.

[3] W. Barrett, E. Mortensen, and D.Taylor. An image space algorithm for morphological contour interpolation. In *Graphics Interface '94*, pages 16–24, 1994.

[4] K.J. Bathe. *Finite Element Procedures in Engineering Analysis*. McGraw-Hill, Prentice-Hall, NJ, 1982.

[5] M.I.G. Bloor and M.J. Wilson. Local control of surfaces generated using partial differential equations. *Computer Graphics*, 18(2):161–169, 1994.

[6] J.D. Boissonnat. Shape reconstruction from planar cross sections. *Computer Vision, Graph. Image Proc.*, 44(1):1–29, 1988.

[7] H.N. Christiansen and T.W. Sederberg. Conversion of complex contour line definitions into polygonal element mosaics. *Computer Graphics*, 12(2):187–192, 1978.

[8] V. Cingoski, M. Ichinose, K. Kaneda, and H. Yamashita. Analytical calculation of magnetic flux lines in 3d space. *IEEE Trans. on Magnetics*, 30(5):2912–2915, 1994.

[9] A.B. Ekoule, F.C. Peyrin, and C.J. Odet. A triangulation algorithm from arbitrary shaped multiple planner contours. *ACM Trans. on Graphics*, 10(2):182–199, 1991.

[10] H. Fuchs, Z.M. Kedem, and S.P. Uselton. Optimal surface reconstruction from planer contours. *Comm ACM*, 10:693–702, October 1977.

[11] K. Kaneda, F. Kato, E. Nakamae, T. Nishita, and T. Noguchi. Three dimensional terrain modeling and display for environment assessment. *Computer Graphics*, 23(3):207–214, 1989.

[12] E. Keppel. Approximating complex surface by triangulation of contour lines. *IBM J. Res. Dev.*, 19:2–11, January. 1975.

[13] K. Komatsu, Y. Shinagawa, T. Kunii, and M. Ueda. Terrain shape reconstruction from contours based on shrinking deformation. *Trans. of IEICE, D-II(in Japanese),* 79(6): 1072-1079, 1996.

[14] J.S. Lee and S.J. Chung. Reconstruction of 3d terrain data from contour map. In *MVA'94 IAPR Workshop on Machine Vision Applications,* pages 28 l-284, 1994.

[15] M. Lounsbery, C. Loop, S. Mann, D. Mayers, J. Painter, T. Derose, and K. Sloan. A testbed for the comparison of parametric surface methods. In *SPIE/SPSE symposium on electronic imaging science and technology(Santa Clara, Calif., Feb.),* 1990.

[16] D. Meyers, S. Skinner, and K. Sloan. Surface from contours. *ACM Trans. on Graphics,* 11(3):228–258, 1992.

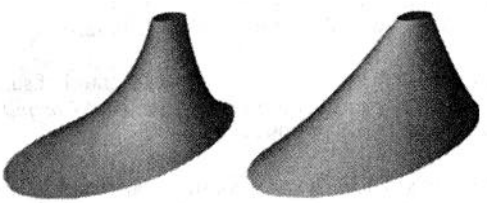
[17] G.M. Nielson. A method for interpolating scattered data based upon a minimum norm network. *Amer. Math. Soc.Math.Comput.*, 40:253–271, 1983.

[18] H. Park and K. Kim. 3-d shape reconstruction from 2-d cross-sections. *Journal of design andmanufacturing,* 5(3): 171-185, 1995.

[19] J. Peter. Smooth mesh interpolation of a mesh of curves. *Computer Aided Geometric Design,* 22:109–120, 1990.

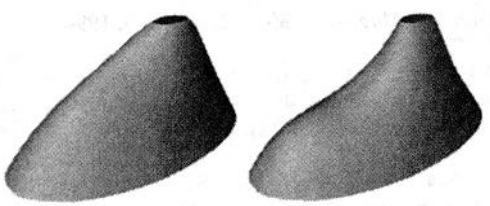
[20] Y.F. Wang and J.K. Aggarwal. Surface reconstruction and representation of 3d scenes. *Pattern recognit.,* 19(3): 197-207, 1986.

[21] O.C. Zienkiewicz and R.L. Taylor. *The Finite Element Method.* McGraw-Hill, London, 1988.



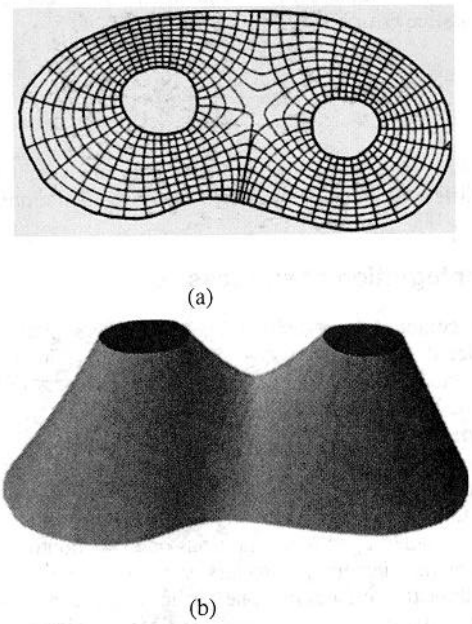
(a) (b)

Figure 6: Difference between free-controlled and gradient controlled PDE surfaces. (a) a free controlled PDE surface. (b) a gradient controlled PDE surface.



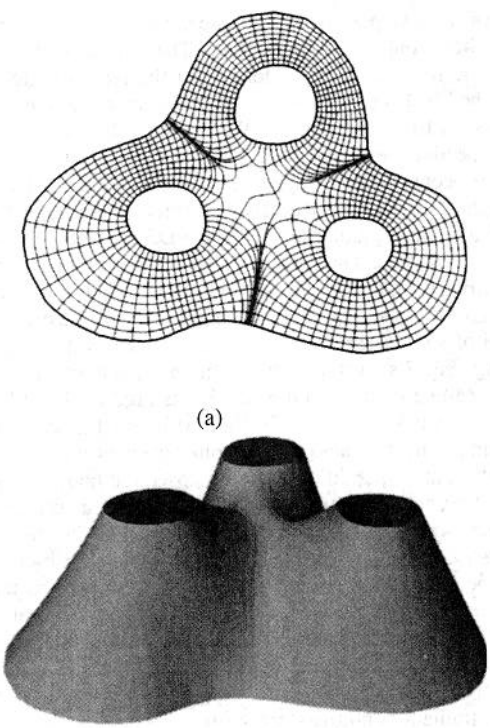
(a) (b)

Figure 7: Effects of gradient boundary conditions on gradient controlled PDE surfaces.



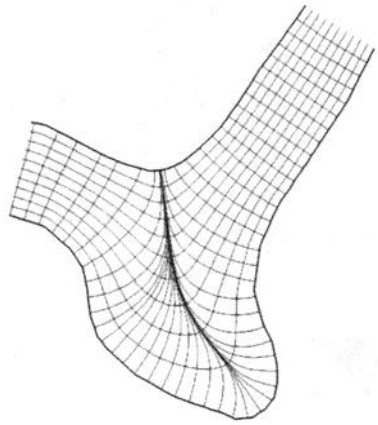
(a) (b)

Figure 8: A surface with two branches. (a) interpolated contours and slope lines. (b) perspective view displayed by polygonal patches.

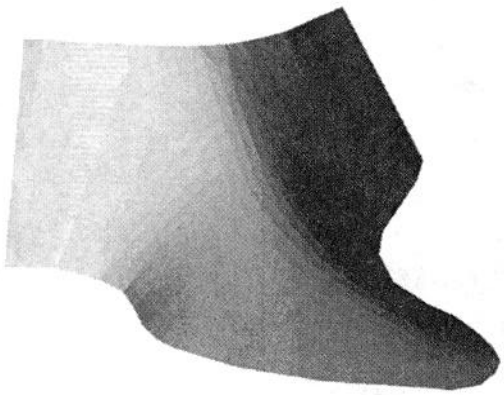


(a) (b)

Figure 9: A surface with three branches. (a) interpolated contours and slope line(b) perspective view displayed by polygonal patches.

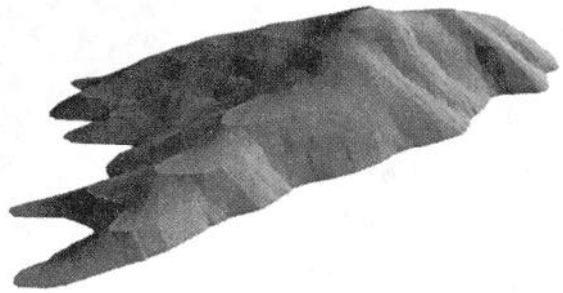


(a)



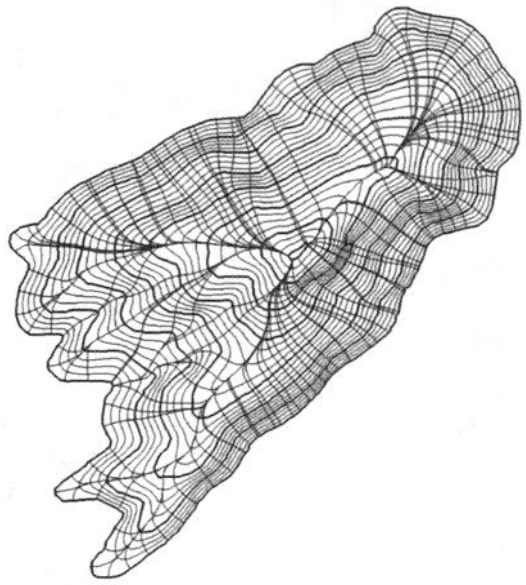
(b)

Figure 10: A surface reconstructed from two contours with very different shapes from each other. (a) interpolated contours and slope lines. (b) perspective view displayed by polygonal patches.



(a)

(b)



(c)

Figure 11: A surface reconstructed from a part of a real map. (a) the surface reconstructed without consideration of smoothness across the contours. (b) the surface reconstructed by the proposed method. (c) the interpolated contours and slope lines of the surface in (b).