

## Construction of Branching Surface from 2-D Contours

**Kailash Jha**

Assistant Professor, Faculty-in-Charge, CAD/CAM Center Dept. of Mechanical Engineering & Mining Machinery Engineering  
Indian School of Mines University, Dhaband, India

**Abstract -** In the present work, an attempt has been made to construct branching surface from 2-D contours, which are given at different layers and may have branches. If a layer having more than one contour and corresponds to contour at adjacent layers, then it is termed as branching problem and approximated by adding additional points in between the layers. Firstly, the branching problem is converted to single contour case in which there is no branching at any layer and the final branching surface is obtained by skinning. Contours are constructed from the given input points at different layers by energy-based B-Spline approximation. 3-D curves are constructed after adding additional points into the contour points for all the layers having branching problem by using energy-based B-Spline formulation. Final 3-D surface is obtained by skinning 3-D curves and 2-D contours. There are three types of branching problems: (a) One-to-one, (b) One-to-many and (c) Many-to-many. One-to-one problem has been done by plethora of researchers based on minimizations of twist and curvature and different tiling techniques. One-to-many problem is the one in which at least one plane must have more than one contour and have correspondence with the contour at adjacent layers. Many-to-many problem is stated as  $m$  contours at  $i$ -th layer and  $n$  contours at  $(i+1)$ th layer. This problem can be solved by combining one-to-many branching methodology. Branching problem is very important in CAD, medical imaging and geographical information system(GIS)

**Keywords:** Surface reconstruction, Branching surfaces, skinning, surface approximation.

### 1. Introduction

Construction of 3D surface from 2D contours is very important for CAD (Rapid prototyping, NC machining), Medical imaging and Geographical Information System. Technologies such as magnetic resonance imaging (MRI), computed topography (CT), and ultrasound imaging allow measurements of internal properties of objects to be obtained in a nondestructive fashion. These measurements are usually obtained one slice at a time, where each slice is a 2D array of scalar values corresponding to measurements distributed over a plane passing through the object. The set of planes generating the slices are usually parallel to each other and may be equi-spaced along some axis through the object.

Once these slices have been obtained, the goal is to enable a human to easily visualize, in 3D, this large collection of data. Many algorithms have been developed for this purpose, but they can all be classified into two categories, Volume Rendering Methods and Surface Reconstruction Methods. Volume rendering is used to show the characteristics of interior of the solid. In surface rendering a geometrical representation is used to model the object or structure to be visualized based on original data such as edge, mesh, polygon, triangle or pixel. The present

work focuses on surface reconstruction method. Generation of 3D surface from 2D contours has four basic steps: (a) Correspondence, (b) Tiling, (c) Branching and (d) Generation of surface.

The correspondence problem involves finding the correct connections between the contours of adjacent slices. Fig. 1 shows an example with four different joint topologies (b)-(e) resulting from the same cross sections as in (a). If the distance between slices is large, a *prior* knowledge or global information is required to determine the correct correspondence.

(b) Tiling means using slice chords to triangulate the strip lying between contours of two adjacent slices into tiling triangles. A slice chord connects a vertex of a given contour to a vertex of the contour in an adjacent slice. Each tiling triangle consists of exactly two slice chords and one contour segment.

In the current work focus is given on branching problems. The contours given have known correspondence. Our approach does not require tiling because it is based on skinning of compatible contours.

(c) A branching problem occurs when a contour in a layer corresponds to more than one contour in an adjacent slice. Fig. 2a shows that contour C3 of slice S2 branches into C1 and C2 of slice S1

A contour in one slice having no corresponding contour in an adjacent slice forms either a hole or the beginning/end of a vertical feature. The possibility of branching significantly complicates the task of tiling. It creates the problem of branching surface reconstruction.

\*Corresponding author:

Tel: +91-326-223-5436(O)/ +91-326-2235536®

Fax: +91-326-221-0028

Homepage: <http://www.ismdhanbad.ac.in/depart/memme/faculty.htm#8>

E-mail: kailash\_jha@hotmail.com

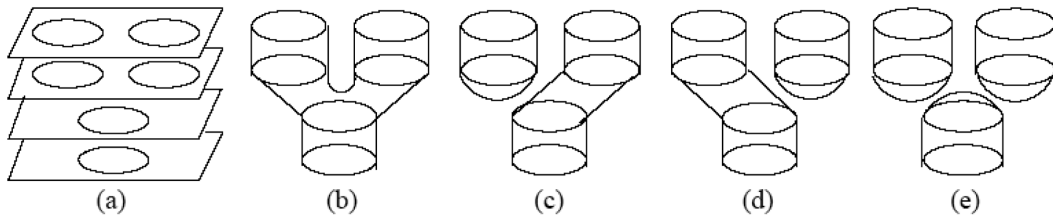


Fig. 1. Correspondence problem (a) cross section contour (b)-(e) different topologies with the same cross section as in (a).

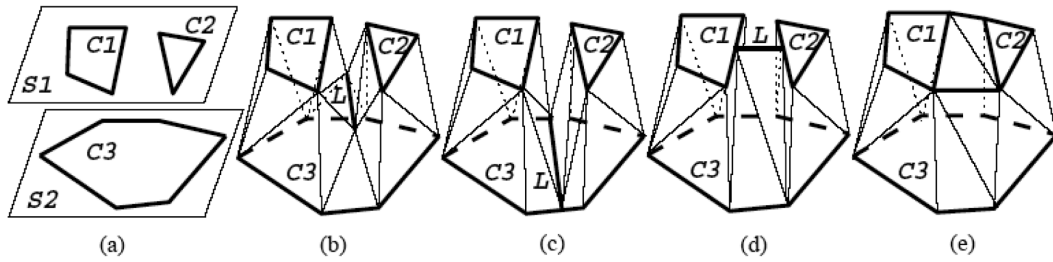


Fig. 2. Different reconstruction for branching contours: (a) Branching contours in adjacent slices; (b)-(e) different surface constructions.

Branching regions can be modeled by interpolating many intermediate contours. This method generates a smooth surface at the cost of a larger number of triangles. Other branch processing approaches can be classified into four methods shown in Figs.2b - 2e. Fig. 2b shows that a curve L or a point is added between two slices to model the valley or saddle point formed by the branching. The added curve L is placed at slice S2 in (c). One or more line segments are added to form a composite contour as in (d); thereafter the tiling between the composite contour and C3 becomes one-to-one. Fig. 2e also forms a composite contour, which is the convex hull of the branching contours, in order to have one-to-one tiling. The branching region between C1 and C2 is filled up by horizontal triangles. In terms of topological correctness, the best branching handling is (b) because it corresponds to the expected physical object better than the others do. We can dip down the middle of the bridge to model the saddle point of the branching region. This approach works well only in simple branching cases based on tiling.

Current branching problems has been solved for known correspondence and starting points for the contours at different layers. A complex 3D contour is generated for layer having branching problem, which converts many-to-many branching problems into one-to-one branching problem.

In the present work, linear merging of geometrical centers of contours is done to get 3-D curves. It is assumed that the contours and their geometrical centers are merging linear way to contours and their geometrical centers at adjacent layer for branching. For example, two contours C1 and C2 are shown in Fig. 3 at layer i having geometrical centers at g1 and g2 respectively, which are merging linearly to the geometrical center (g3) of adjacent contour C3 at (i+1)th layer.  $\underline{ad}$  and  $\underline{bd}$  are parallel to  $\underline{g1g3}$  and  $\underline{g2g3}$  respectively to give 3-D curve, which is comprised of  $\underline{db}$ , C2,  $\underline{bd}$ ,  $\underline{da}$ , C1 and  $\underline{ad}$  respectively. Once 3-D curve is obtained and the remaining contours at other layers do not have any branches, then the contours are skinned to get final 3-D surface otherwise same process is repeated. Fig. 3a shows the

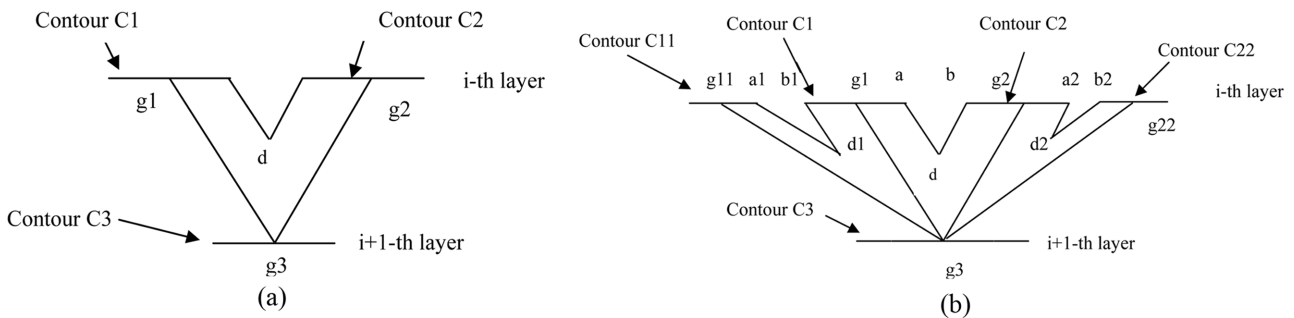


Fig. 3. (a) Illustrated example to convert contours to a 3-D composite contour for a layer having two contours, (b) Illustrated example to convert contours to a 3-D composite contour for a layer having four contours.

contour points as well as other additional points. In Figure 3b another example is shown which have four contours (C11, C1, C2, C22) at  $i$ -th layer, which are merging to a contour (C3) at  $(i+1)$ th layer. In this example there are three “d”(d1, d and d2) and the 3D composite curves are obtained similarly as in example shown in Fig 3a. Center of the contours C11, C1, C2, C22 and C3 are  $g_{11}$ ,  $g_1$ ,  $g_2$ ,  $g_{11}$  and  $g_3$  respectively. The final 3D curve is **Fig 3a** Illustrated example to convert contours to a 3-D composite contour for a layer having two contours **Fig. 3b** Illustrated example to convert contours to a 3-D composite contour for a layer having four contours comprised of db, part of C2, a2d2, d2b2, C22, b2d2, d2a2, rest part of C2, bd, da, part of C1, b1d1, d1a1, C22, a1d1, d1b1, part of C1 and ad.

The remainder of the paper is organized as follows. Previous work is presented in section 2. Current work is described in section 3. Results and discussions are included in section 4. Conclusions are presented in section 5.

## 2. Previous Work

The literature that is devoted to many-to-one problem can be classified into four main families. The family of contour connection methods attempted to artificially render one-to-many problem into a one-to-one by connecting the disjoint contour with line [1] or triangulate facet bridge [2]. The first choice is applicable for simple cases, while the second constrains unnaturally the saddle points of the branching surface to lie on the plane containing the disjoint contours.

The second family is based on introduction of intermediate contour, which splits the original problem into two problems one-to-one and a new one-to-many. The second problem is further simplified into  $m$  one-to-one problems. This idea has been proposed in [3, 4] and has been implemented in [5].

The family of partial contour connection and hole filling has been proposed in references [6-8] is characterized by matching partially the disjoint contours with the single contours of the neighboring plane, thus leaving a number of whole which are finally filled in final step. In [9], where the case one-to-two is being treated, the single hole is filled by an appropriate chosen hyperboloid.

Finally, the family of implicit schemes relies on the assumption that we possess implicit representation of contours composing the cross-sections. Then an implicit interpolant can be obtained by taking a convex combination of contour representation [10], or implying the distance function [5, 11].

Current work is based on construction of 3D curves for the contours at a layer having correspondence with contour at adjacent layer. The present technique is simple and number of step involved is less, only few extra points are added in given inputs once there is branching problem. In this technique only the composite 3D curves is generated wherever there is

branching problems and the final surface is obtained by skinning the curves which is free from branching problem. In traditional technique [1, 6] to solve the branching problem is comparative complex because of large number of triangles and numbers of steps used, hence the computationally costlier. Even latest technique to solve branching problem based on skinning, trimming and hole filling seems costly and results shifting of joining section toward the layer having single contour. Tiling techniques [1, 2] result  $C^0$  triangulated surface. In the present work linear portions of the 3D curve are maintained by imposing constraints.

## 3. Current work

### 3.1 Construction of 3D curves for contours having correspondence

In the current work, the contours of the layer having branching are first converted into single contour (3D curve). In this research, approximation technique is used because the layer having 3D curve have a large number of contour points due to merging of several contours. In present formulation getting compatible curves is difficult. In the current work, construction of 3D surface has been performed for the contours at different layers may have branching problems. First, the layer having more than one contours have been identified. The next task is to convert these contours into a single 3D curve, which will be skinned later. The geometrical center for these contours is determined first, the contours at adjacent layers are identified which have branching problems. The geometrical center of contours at adjacent layer is determined. In Fig. 3a, C1 and C2 are the contours at  $i$ -th layer having correspondence with contour C3 at  $(i+1)$  th layer. The geometric center of these contour are  $g_1$ ,  $g_2$  and  $g_3$  respectively. A tangent is drawn from the right most point ‘a’ of contour C1 parallel to g1g2. Similar tangent is drawn parallel to g2g3 from left most point ‘b’ of the contours C2. These tangents intersect at point ‘d’. A three-dimensional curve will be constructed by C1, ad, db C2, bd, ad and C2. This curve converts the multiple contours into single contour (3-D curve). In Fig 3a, only two contours are present which have correspondence with contour C3 at adjacent layer. If more contours are available as in case of an example shown in Fig 3b for correspondence then the same technique is repeated to get the 3D curve. Once the 3D curves are constructed. If all the layer having the single contour (curve) then all the contours are skinned to get three-dimensional surface.

Getting skinned surface from the constructed curves is difficult because the curves are not compatible. In these circumstances the final compatible curves are obtained by approximation technique instead of interpolation.

### 3.2 Algorithm of Approximation

In this technique, a compromise between accuracy

and the cost has been done. Cost deals with the storage cost as well as the computational cost. The present algorithm starts with input points, which are given for each contour at a layer. Maximum number of input points ( $m + 1$ ) among the 3D curves is determined. We assume that maximum number of common control points will not be more than this number ( $m+1$ ). Common degree ( $p$ ) and accuracy are also inputs for this algorithm. The overall degree of the curves is user-defined input. If curves are given as input instead of the points they are derived from the different curves. Input curves may be of different degree. Current algorithm is based on binary search to get optimal number of control points for given accuracy (see references [12, 13]. Each sectional curves are verified by energy based B-Spline interpolation technique which is described in next section for given accuracy.

**3.3 Energy based curve formulation**

Approximation of contour and 3D curves are obtained by solving the following energy equation:

$$E = \int_t \alpha(C_t(t).C_t(t))dt + \int_t \beta(C_{tt}(t).C_{tt}(t))dt + \int_t \gamma(C_{ttt}(t).C_{ttt}(t))dt + \sum_{i=0}^m \|P_i - C(t_i)\|^2 \tag{1}$$

Where  $\alpha$ ,  $\beta$  and  $\gamma$  are non-negative values called stretching, bending and fitting co-efficient respectively.

A new non-negative coefficient of energy responsible for having smooth curvature is  $\gamma$

In this technique equation (1) is solved.

Approximated cubic B-Spline curve is  $C(t)$

Input points are denoted by  $P_i$

First, second and third derivations of B-Spline curve are  $C_t$ ,  $C_{tt}$  and  $C_{ttt}$  respectively.

Parametric variable is  $t$ . These energy terms have been explained in [12-16]. The squared first and second terms in equation (1) give strain energy contribution due to stretching and bending of the curve respectively. Squared third derivative part of the equation does not have any physical significance. It has some geometrical meaning. The magnitude of the third derivative is a rough estimate of the rate of change of curvature with respect to the parametric value. The fourth term of energy equation indicates the energy stored in spring due to error.

Effects of these coefficients on the approximated curves have been studied in the previous work [13]. Quadratic functional minimization technique has been used in this work, which results in a curve by solving the linear set of equations [14]. Equation (1) has been solved by assuming a B-Spline curve satisfying it. Set of linear equations has been obtained finally. Control points of B-Spline curve are obtained by solving linear equations. If equation (2) is the B-Spline solution for

equation (1)

$$C(t) = \Sigma NX^T \tag{2}$$

Where  $N$  represents B-Spline basis functions and  $X$  represents Control points (12-15). Equation (1) can be rewritten as

$$\begin{aligned} & X^T \left( \alpha \int_t N_t N_t^T dt + \beta \int_t N_{tt} N_{tt}^T dt + \gamma \int_t N_{ttt} N_{ttt}^T dt \right) \\ & X + \gamma_1 (AX - P)^T (AX - P) \tag{3} \\ \Rightarrow & X^T (\alpha K_1 + \beta K_2 + \gamma K_3) X + \gamma (AX - P)^T (AX - P) \\ \Rightarrow & X^T K_L X + \gamma_1 (AX - P)^T (AX - P) \end{aligned}$$

Where

$$K_1 = \int_t N_t N_t^T dt$$

$$K_2 = \int_t N_{tt} N_{tt}^T dt$$

$$K_3 = \int_t N_{ttt} N_{ttt}^T dt$$

$N_t$ ,  $N_{tt}$ ,  $N_{ttt}$  represents first, second and third derivatives of Basis Function.  $K_1$ ,  $K_2$  and  $K_3$  are 4 x 4 matrices [14].  $A$  and  $P$  represent co-efficient matrix and input points respectively.

$$K_L = \alpha K_1 + \beta K_2 + \gamma K_3$$

If the basis function assumed is uniform or in Bezier form  $K_L$  can be calculated analytically. In case of non-uniform basis function numerical technique (Gaussian quadrature) can be used to get  $K_L$  matrix. Constraints are given by

$$C_k X = D_k$$

Where  $k$ , varies from 0 to (number of constraints-1).

This equation may contain the positions, tangents and higher derivative data.

The final matrix equations can be expressed as reference [12].

$$\begin{pmatrix} K + \gamma_1 A^T A & C^T \\ C & \phi \end{pmatrix} \begin{pmatrix} X \\ V \end{pmatrix} = \begin{pmatrix} \gamma_1 A^T P \\ D \end{pmatrix} \tag{5}$$

Where  $V$  is a vector storing Langrange Multiplier and  $X$  are the unknown control points.

$K$  is the assembled stiffness matrix with size  $(n + 1) \times (n+1)$ .

$C$  is  $n$ -constraints  $\times (n + 1)$  matrix.  
 $n$ -constraints indicates number of constraints given.  
 Above matrix equations have been solved by Gauss-elimination to get control points.  
 Details of this formulation are available in [12-15].

**3.4 Surface Skinning**

Skinning is a process of blending the sectional curves together to form a surface (see references [5]). Blending direction is in  $v$ -direction, which is also known as longitudinal direction. Control points of contours obtained by energy-based approximation are compatible, which are skinned by using equation (7)

$$S(u, v) = \sum_{i=0}^m \sum_{j=0}^n V_{ij} M_{j,q}(v) N_{i,p}(u) \tag{7}$$

Control points ( $V_{ij}$ ) for  $i = 0, 1, 2, \dots, m$  and  $j = 0, 1, 2, \dots, n$  are obtained by energy based B-Spline approximation in  $u$ -direction. Final control nets are obtained by interpolating or approximating  $V_{ij}$  (compatible control points) in  $v$ -direction. All  $j$  column of  $V_{ij}$  are taken for interpolation in  $v$ -direction. Chord-length approximation has been used to get the approximate parametric value of control points for skinning.

**4. Results and Discussions**

Several data have been tested. Results have been displayed by using OpenGL on a PC using windows operating system. Zooming, panning and rotation have been implemented for interactive visualization of models having branches. The programming language used in present formulation is C++. Fig. 4 shows the contour points as well as other preprocessed points. Figs. 5 and 6 show the 3-D constructed surface and its shaded view obtained by current implementation.

Fig. 7 shows the contour points for four layers in which first layer have three contours having correspondence

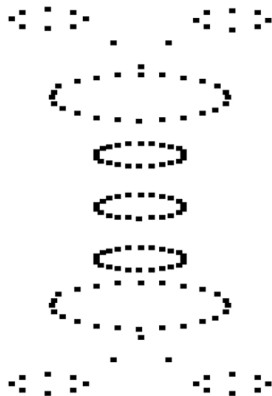


Fig. 4. Preprocessed input points.

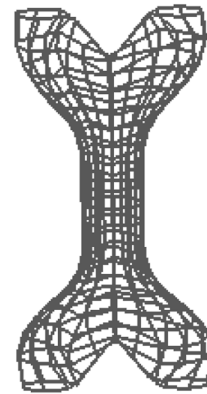


Fig. 5. 3D surface from contours at 7 layers.

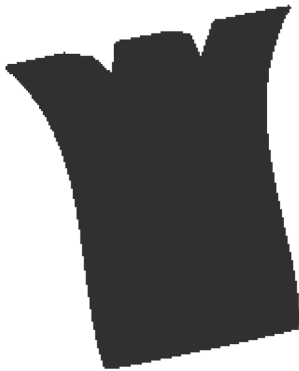


Fig. 6. 3D shaded surface

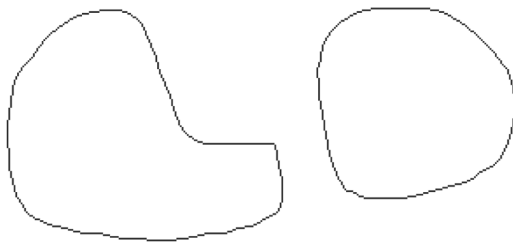


Fig. 7. Layer having three contours showing correspondence.

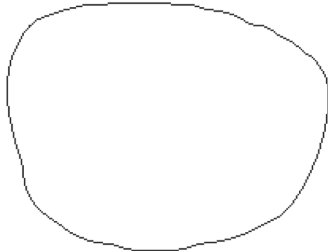
with the contour of adjacent layer. Fig. 8 shows the shaded view of the constructed surface whose contours have been shown at different layers in Fig 7. In the present formulation, 3D curves are approximated by keeping  $\alpha, \beta$  and  $\gamma$  value equal to zero. It is like a least square B-Spline fitting because curves having sharp corners. Curves at the other layers are approximated by energy based B-Spline formulation. Fig 9 and Fig 10 show the contours at top layer and bottom layer respectively. The model with intermediate contours is



**Fig. 8.** Shaded view of final surface.

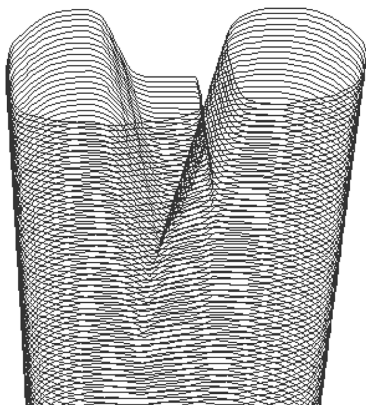


**Fig. 9.** Top contours.



**Fig. 10.** Bottom contour.

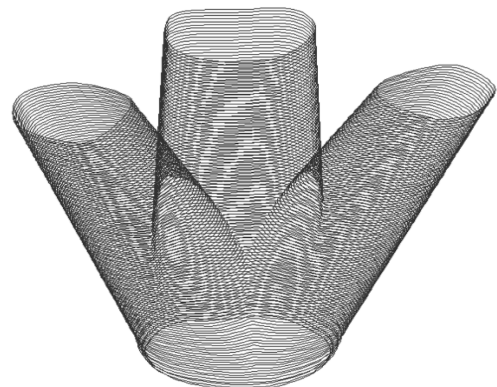
shown in Fig. 11. Fig. 12 shows some other model with contours.



**Fig. 11.** Model with contours shown in Figs. 9 and 10.

It has been found in this research that taking two contours and getting ruled surface out of them may be an effective technique to get branching surface. In this case the branching surface will be constructed segment by segment. A segment comprises two layers, which may have branching problems. Resolution of branching is same as done in this work. A brief overview of branching problems has been explained in reference [17], which involves skinning, trimming and hole filling. The results given in [17], the joining sections are shifted toward the layer having single contour.

Figure 13 shows points of a model of a crack surface. Figure 14 shows the control polygons obtained by the algorithm explained in section 3.2. Control net and skinned surface are shown in Figure 16. Figure 17 shows another model having points and control polygons. Figure 18 shows the skinned surface for the points shown in Figure 17. Figure 19 shows the control polygon obtained by the algorithm given in section 3.2 for the points shown in Figure 7. The building direction is



**Fig. 12.** Model with contours.



**Fig. 13.** Points.



Fig. 14. Control polygons according to section 3.2.



Fig. 15. Control net.

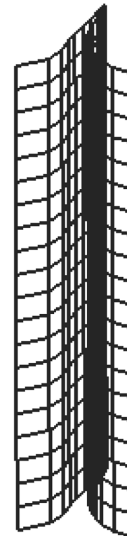


Fig. 16. Surface.



Fig. 17. Points and control polygons.

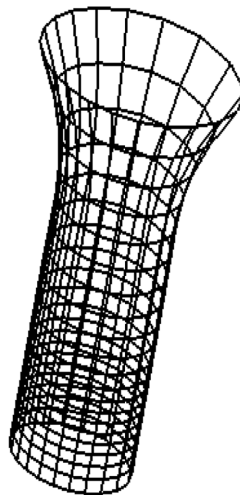


Fig. 18. skinned surface.

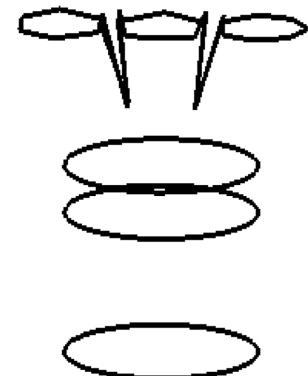


Fig. 19. Control polygon for the model shown in Fig. 7.

important for correct skinned surface. It should be consistent between contours of adjacent layers.

## 5. Conclusions

In the present work, an attempt has been made to construct branching surface from 2-D contours given at different layers having branching problem. Contours are given as points at different layers. 3D curves have been obtained for given contours and their correspondence contour at different layers. The surface has been obtained by skinning of compatible curves. Compatible curves at different layers are obtained by approximation technique. Segment-by-segment design based on rule surface may give better results.

## Acknowledgement

Author is grateful to Late Shri Harischandra Jha for his motivation toward the research and development work. His is no more but his inspiration is always with me to do the novel work.

## References

- [1] Christiansen, H N and Sederberg, T. W., (1978), Conversion of complex contour line definitions into polygonal element mosaics, *Computer Graphics (SIGGRAPH'78 Proceedings)*, **12**, 187-192.
- [2] D Meyers, S. Skinner and K. Sloan. (1992), Surfaces from contours. *ACM Trans. On Graphics*, **11**, 228-259.
- [3] Y. Shinagawa and T. L. Kunii. (1991), The homotopy model: *A generalized model for smooth surface generation from cross sectional data. Visual Computer*, **7**,

- 72-86.
- [4] A. B. Ekole, F. C. Pysin and C. L. Odet. A. (1991) triangulation algorithm from arbitrary shaped multiple planner contours. *ACM trans. on Graphics*, **10**, 182-199.
- [5] J. Jeong, K. Kim, H. Park, H. Cho, and M. Jung. (1999), B-Spline surface approximation to cross section using distance maps. *Adv. Manuf. Techn*, **15**, 876-885.
- [6] C. L. Bajaj, E. J. Coyle, and K. N. Lin. (1996), Arbitrary topology shape reconstruction from planner cross sections. *Graphical Models & Image Proc.*, **58**, 524-543.
- [7] G. Barequet, D. Shapiro, and A. Tal. (2000), Multilevel sensitive reconstruction of polyhedral surfaces from parallel slices. *Visual Computer*, **16**, 116-133.
- [8] G. Barequet and M. Sharir. (1996), Piecewise-linear interpolation between polygonal slices. *Comp. Vision & Image Underst.* **63**, 251-272.
- [9] T. N. T. Goodman, B. H. Ong, and K. Unsworth. (1993), Reconstruction of  $C^1$  closed surfaces with branching. In G. Farin, H. Hagen, and H. Noltemeier, editors, *Geometric Modelling*, pages 101-115. Springer, Vienna.
- [10] S. Bedi. (1992), Surface design using functional blending. *CAD*, **24**, 505-511.
- [11] M. S. Floater and G. Westgaard. (1996), Smooth surface reconstruction from cross-sections using implicit surfaces. Report STF42, Sintef.
- [12] Park, H., Kim, K., Lee, Sang-Chan. (2000) A method for approximate NURBS curve compatibility based on multiple curve refitting, *Computer Aided Design*. **32**(20), 237-252.
- [13] Jha, Kailash. (2004) Minimum Energy Based Interpolation for Interactive Incremental Well-Path Design, *Engineering Computations*, **21**(7), PP 67-69.
- [14] Jha, Kailash. (2005), Energy based multiple refitting for skinning, *International Journal of CAD/CAM*, **5**(1), 11-18.
- [15] Fang, L., Gossard, D. C. (1995), Multidimensional Curve Fitting to Unorganized Data Points by Non-linear Minimization, *Computer Aided Design*. **27**(1), 45-58.
- [16] Park, H., Kim, K. (1996), Smooth Surface Approximation to Serial Cross-Section, *Computer Aided Design*, **28**(12), 995-1005.
- [17] Gabrielides, N. C., Ginnis, A. I., Kaklis, P. D. (2006), Constructing Smooth Branching Surfaces from Cross Sections, ACM Symposium on Solid and Physical Modeling, Cardiff, Wales, UK, June 06-08, Pages 161-70.

---

**Dr. Kailash Jha** is currently Assistant Professor in Dept. of Mechanical Engineering & Mining Machinery Engineering, Indian School of Mines University, Dhanbad, Jharkhand, India. He did his schooling from BDAH School, Nanour (Madhubani). He received his bachelor and master degrees in Mechanical Engineering from REC (NIT), Durgapur (WB) in 1991 and 1993 respectively. He received his Ph D in Mechanical Engineering from Indian Institute of Science, Bangalore, India. His research interests are soft computing, Curve and surface modelling, Feature based design and Geometrical reasoning.

---



Dr. Kailash Jha