芸術科学会論文誌 Vol. 10, No. 1, pp. 12 – 27

Curve Mesh Modeling Method of Trimmed Surfaces for Direct Modeling

Yuta Muraki∗1 **Kouichi Konno**∗2 **Yoshimasa Tokuyama**∗3

∗1∗2Faculty of Engineering, Iwate Univ. ∗3Faculty of Engineering, Tokyo Polytechnic Univ.
E-mail: murakiyuta@lk.cis.iwate-u.ac.jp

Abstract
In the shape modeling with 3D CAD systems, the trimmed surface is quite popular. For instance, Japanese Industrial Standards (JIS) models contain a lot of notches, expressed using trimmed surfaces. Since trimmed surfaces are directly modified in direct modeling, it has a big restriction in the shape modification. It is effective to apply a new free-form surface to a closed region composed of the modified edges because the consistency of a trimmed surface can be maintained. This paper proposes the method of fitting a free-form surface by using the offset curve. To be more concrete, an offset curve is generated according to the tangent planes and a point cloud is generated. After that, a B-spline surface is generated using the generated point clouds and the boundary curves, so that a new trimmed surface is generated. Our method is effective for direct modeling that directly modifies the boundary edges of the trimmed surface representing a notch shape.

Keywords : Direct Modeling, Trimmed Surface, Surface Fitting, Offset Curve, Notch Shape

1 Introduction

In the shape modeling operations with 3D CAD systems, the trimmed surface[1] is quite popular. In the shape modeling with CAD systems, there is a design method called “ feature base modeling ”. Since CAD systems define their unique features, data might not be converted correctly between different CAD systems. Due to this restriction, the converted shape data must be modified in order to correct it. The boundary edge that trims a surface is the curve lying on the surface within a certain tolerance. If the surface shape and the trimming boundary edge are modified separately, the consistency of them will collapse. Whenever the control points of the surface are moved or the shape of the boundary edge is modified, the consistency of the trimmed surface must be maintained, and this is a big restriction in the modeling operations.

If the shape of an edge is modified using the direct modeling, the control points of the surface must be moved appropriately so that the resultant edge must lie on the surface within a certain tolerance. In general, the sequence of the boundary edges of a trimmed surface forms an N-sided closed region. Once the shape of the boundary edges is separately modified, the control points of the surface cannot be moved easily in order to lay the boundary edges on the surface within a certain tolerance. Therefore, after direct modeling operations, it is effective to apply a new free-form surface on the closed region so that the geometrical consistency can be maintained.

In general, the JIS 3D models often contain notches, as shown in Figure 1. Figure 1 (b) shows one of the trimmed surfaces of the model (a) and its control points. As the control points shows, the notch trims the original surface.

The methods of fitting a surface to an N-sided closed region are classified into surface interpolation and N-side filling[3]. The surface interpolation method interpolates a closed region with free-form surfaces so that the boundary edges of the closed region coincide with the boundary curves of the free-form surfaces. The N-side filling method generates a quadrilateral trimmed surface that includes a closed region of the boundary curves. The technique of the surface interpolation method is detailed first. Piegl et al. introduced an interpolation method with the angle tolerance ε to generate smooth surfaces[4]. In his method, the angle of the normal vectors on an arbitrary point of the common boundary edge between adjacent patches becomes smaller than ε. Yi-Jun Yang et al. enhanced the Piegl's method to apply ra-

tional curve meshes[5]. Sederberg et al. proposed T-splines that relaxed the restriction of the topology of NURBS[6]. In his method, the control points that are not important geometrically can be removed. Therefore, T-spline reduces a lot of control points in order to permit T-shaped connection. Chongyang Deng et al. proposed an efficient algorithm for constructing Catmull-Clark surfaces[7][8]. The method of Chongyang Deng et al. models surfaces by repeating element division. The method need not consider the connection of surfaces, and is suitable for interactive construction of free-form surfaces.

The proposed methods[4, 5, 6, 8] are integrated into Catmull-Clark subdivision[7]. In an N-sided region, the center point and division curves are generated so that the region is divided into N quadrilateral regions, and a surface is interpolated to each of the generated regions. For example, Figure 2 shows a nine-sided face. Depending on the shape of the boundary edges of a closed region, the division curves are generated outside the region. In this paper, the surface generated outside the region or the undulating surface is defined as a distorted surface. Moreover, the methods[4, 5, 6] cannot be applied to a region with holes. Garcia et al. proposed the method of fitting a surface to an arbitrary N-sided region by dividing the region into a star-shaped N-sided patch and quadrilateral patches, which can be controlled using parameter f[9]. The method[9], however, is not applied to a shape with holes similarly.

Next the trimmed surface generation method is detailed. Tokuyama et al. proposed the method of fitting a bicubic B-spline surface to an N-sided region[10]. The method[10] applies a bicubic B-spline surface to a quadrilateral region enclosed with free-form boundary curves. When a B-spline surface is generated, four boundary curves are generated first, and according to the tangent plane on each of the boundary curves, point clouds are generated onto the tangent plane outside the quadrilateral region. Then, using the four boundary curves and the generated point clouds, the internal control points are calculated for the B-spline surface. The method[10] can be applied to an arbitrary N-sided region composed of convex vertices. If, however, the method is applied to a region with a concave, the generated surface will be distorted.

In this paper, we propose the method of fitting a free-form surface without distortion to a model, even though the surface includes holes or concave shapes. To be more concrete, the Tokuyama's method[10] is enhanced with the following two operations so that a surface can be generated to a region with holes or concave shapes:

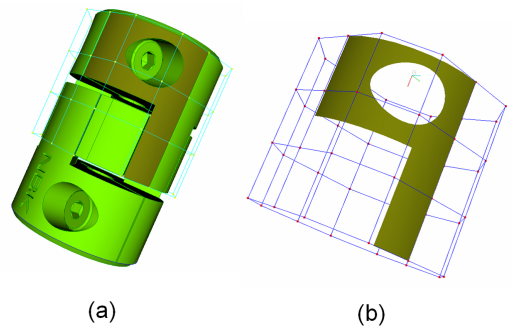


Fig. 1: JIS sample shape model[2]

(1) An offset curve[11] is generated according to the tangent plane around the concave vertex, so that a point cloud is generated.

(2) Cross boundary tangent lines are obtained from the point on the edges except the one connected to the concave vertex, so that the four boundary curves of a B-spline surface are generated.

In our method, boundary edges are input, and a surface is generated so that the boundary edges lie on the surface within a certain tolerance. Therefore, a geometrical consistency is maintained between the surface and the boundary edges. In our method, a bicubic B-spline surface is used since the surfaces used most as the machine parts are cubic. Our method is effective for direct modeling that directly modifies notch geometry.

2 Related Works and Problems

2.1 Surface generation method with two or more surface patches

The method of Garcia et al.[9] inputs boundary edges representing a star-shaped N-sided region and interpolates two or more smooth patches in the region. N regular patches X_n composed of quadrangles

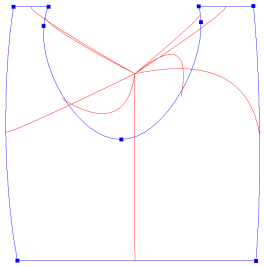


Fig. 2: Example of surface subdivision with Catmull-Clark base

C_i and L_i are generated around the star-shaped N-sided patch. The size of the star-shaped N-sided patch can be controlled using parameter f. If the value of f increases, the N-sided region will be large; and if the value of f decreases, the region will be small. With this method, an intuitive shape operation is possible and an arbitrary N-sided shape can be interpolated with surface patches.

2.2 Method of covering a region with a B-spline surface

Tokuyama et al. proposed the method of fitting a surface by surface fitting method[10]. The method of Tokuyama et al. uses the boundary curves of a B-spline surface that covers an N-sided region and sample points. In the method, a straight line is generated to generate sample points lying on the tangent plane of the boundary edge. In this paper, the straight line is called " a Cross Boundary Tangent Line (CBTL) ".

Suppose a surface is applied to an N-sided region whose boundary edges are drawn in blue in Figure 3 (a). First, as shown in Figure 3 (a), four reference planes are extracted so that they enclose the boundary edges. After that, as shown in Figure 3 (b), the lines are generated based on the tangent plane of the boundary edge, and the intersection points between the straight lines(CBTLs) and the reference planes are calculated. As shown in Figure 3 (c), a sequence of the intersection points is approximated by a B-spline curve, and the boundary curves that cover an N-sided region are generated. After that, sample points are generated on the CBTLs.

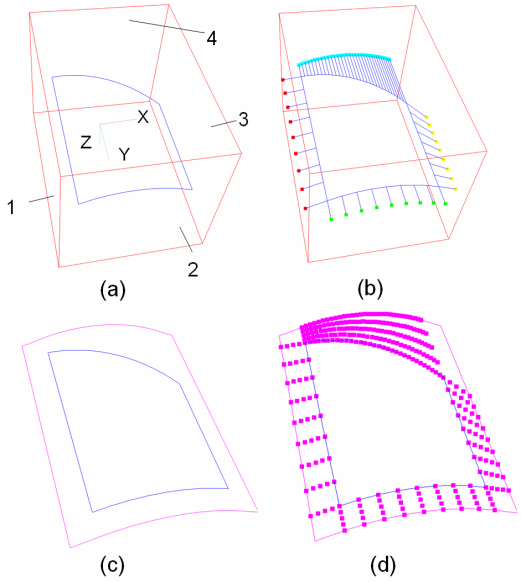


Fig. 3: Calculating sample points from boundary edges

2.3 Problems

The concave shape referred in this paper is defined first. As Figure 4 (a) shows, we pay attention to vertex P. Two edges share vertex P, and the unit tangent vectors on vertex P for the two edges are $\mathbf{v}_1$ and $\mathbf{v}_2$. The outer product vector V is calculated using equation (1). As equation (2) shows, the inner product is calculated between vector V and the mean normal vector $\mathbf{n}$ of the boundary edges of the N-sided region. When the inner product is positive, vertex P is set as a convex vertex, otherwise vertex P is set as a concave one.

$$\mathbf{V} = \mathbf{v_1} \times \mathbf{v_2} \tag{1}$$

$$\mathbf{V} \cdot \mathbf{n} = \begin{cases} > 0 & \text{(convex vertex)} \\ other & \text{(concave vertex)} \end{cases} \tag{2}$$

If the CBTLs generated from a boundary edge are intersecting each other or they are twisted, as shown in the section in the red circle of Figure 4 (b), the boundary edge is regarded as a concave edge. When a shape contains one or more concave vertices or edges defined above, the shape is defined as a concave shape.

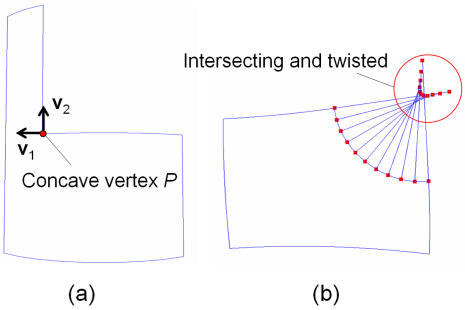


Fig. 4: Defining a concave shape

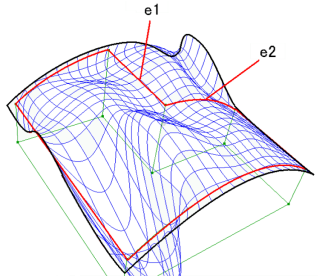


Fig. 5: Example of applying the method of Tokuyama et al. to a concave shape

When the method[9], it is difficult to apply a surface to a concave shape such as an L-shaped region. Moreover, if the method is applied to the region with holes, an N-sided patch cannot be generated and as a result, a surface will not be generated.

If the method[10] is applied to a concave shape, the surface will be generated to undulate as shown in Figure 5. In addition, it is not clear that a surface can be applied to a shape with holes. The reasons why the generated surface undulates are as follows:

(1) The point cloud generated from the CBTLs is generated inside the region around the concave vertex.

As shown in Figure 6 (a), the cross boundary derivatives of the two edges connected to the concave vertex are twisted. Because of this, the CBTLs are generated inside the N-sided region. When a sample point is generated inside the N-sided region, the generated surface undulates since it is restrained to the point cloud that does not lie on the surface. In addition, the boundary curves of the surface may be twisted because of the intersection of the CBTLs.

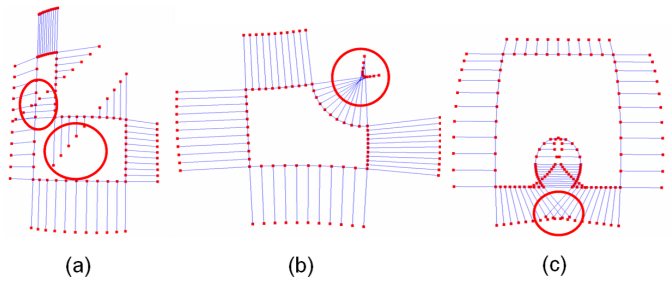


Fig. 6: Generating CBTLs on concave shapes

(2) The method of generating the boundary curves of a B-spline surface is limited and it cannot be applied to concave shapes.

As shown in Figure 6 (b) and (c), the CBTLs generated from a boundary edge of the concave shape may intersect each other or twist. Because of this, if boundary curves are generated from the points obtained from the end points of the CBTLs, the boundary curves will be distorted.

From the reasons described in this section, the method[10] can generate a non-distorted B-spline surface for a convex N-sided region, although a distorted surface may be generated for an N-sided region with a concave shape.

In addition, the posture of a space should be considered. The four reference planes obtained in the method [10] are dependent on the posture of the boundary edges in 3-dimensional space. If the posture of the boundary edges differs, the reference planes will vary, and so will the generated surface.

3 Proposed Methods

3.1 Method of generating four reference planes

The method[10] depends on the posture of a shape in the 3-dimensional space. Due to this, there was a problem that the boundary curve is distorted depending on the posture in 3-dimensional space although the shape was the same. It is necessary to generate four reference planes so that their boundary curves are not distorted. In our method, a local coordinate system is

newly set using shape data and four reference planes are generated using the local coordinate system. The four reference planes are generated in the following procedure:

(1) From the center of the boundary edges of a closed region and vector $\mathbf{n}$ for which the mean normal vector is normalized, the mean plane is generated[14].

(2) The boundary edges are projected to the mean plane and projected boundary edges are generated.

(3) Among the projected boundary edges, the longest edge is selected. The $\mathbf{X}$-axis of the local coordinate system is decided from the start point $\mathbf{p}_0$ and the terminal one $\mathbf{p}_1$ of the longest edge as:

$$\mathbf{X} = \frac{\mathbf{p_1} - \mathbf{p_0}}{|\mathbf{p_1} - \mathbf{p_0}|} \tag{3}$$

(4) The outer product vector between vectors $\mathbf{X}$ and $\mathbf{n}$ is calculated to decide the $\mathbf{Y}$-axis as:

$$\mathbf{Y} = \mathbf{n} \times \mathbf{X} \tag{4}$$

The obtained $\mathbf{X}$ and $\mathbf{Y}$ axes are set on the mean plane.

(5) The unit mean normal vector $\mathbf{n}$ is regarded as the $\mathbf{Z}$-axis. From the obtained $\mathbf{X}$, $\mathbf{Y}$ and $\mathbf{Z}$ axes, the local coordinate system is set.

(6) In the local coordinate system, the bounding box of the projected edges is obtained. From the bounding box and the $\mathbf{Z}$-axis, four reference planes are calculated.

The obtained four reference planes are not dependent on the posture of a shape in the 3-dimensional space; i.e., the reference planes are obtained using affine invariable. As a result, the problem where the distortion of the boundary curves is generated in the existing method is improved.

3.2 New method of generating CBTLs

The CBTL in the method[10] represents the tangent plane at a point on the boundary edge. In order to apply a surface to a region, after the CBTLs are generated, a point cloud is generated outside an N-sided region according to each of the CBTLs. Then, surface fitting is performed according to the point cloud, and a B-spline surface is generated. Around a concave shape, a point cloud is generated inside the N-sided region. In addition, if the CBTLs intersect or twist each other, a point cloud scatters to the space. From these reasons, the point cloud located on the tangent planes will be vague. In other words, some points are generated on a surface where they should not exist and a surface will be applied vaguely to a region.

To solve the above problems, our method enhances the CBTL generation method around concave vertex.

To be more concrete, the method [10] is extended according to the operations below, so that the problem described in section 2.3 will be solved.

(1) If one of the end points of a boundary edge is a concave vertex, as shown in Figure 6 (a), the CBTLs are generated from an offset curve based on the tangent planes. The offset curve generation is described in section 3.3.

(2) If a boundary edge is a concave edge, as shown in 6 (b), the CBTLs are generated from the offset curve, in the same way as (1).

(3) If two or more CBTLs of a boundary edge intersect, as shown in Figure 6 (c), the CBTLs are generated again in the direction based on curves the tangents of the edge at sample points.

As shown in Figure 7, the outer product is obtained using equation (5) between the normal vector $\mathbf{a}$ and tangent vector $\mathbf{b}$ of the tangent plane at a sample point of the boundary edge. The obtained vector $\mathbf{c}$ is set as the direction of the CBTLs. Then, the intersection points are obtained between the direction vector of the CBTLs and the four reference planes, so that new CBTLs are generated between the nearest intersection point and the sample point.

$$\mathbf{c} = \mathbf{a} \times \mathbf{b} \tag{5}$$

The operations described above are applied to a concave shape shown in Figure 6, and the CBTLs are regenerated as shown in Figure 8.

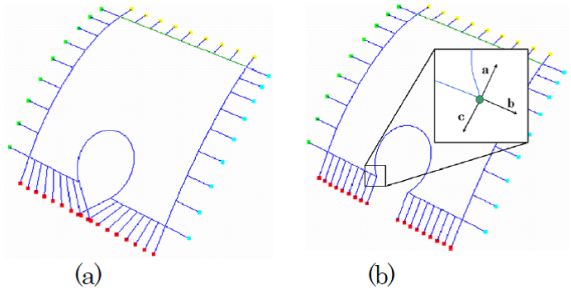


Fig. 7: Regenerating CBTLs

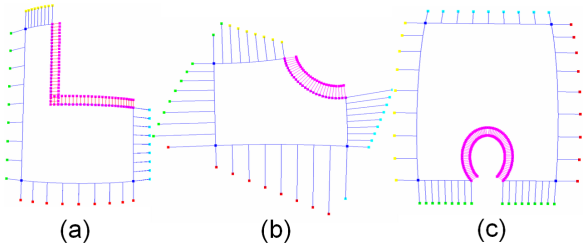


Fig. 8: Result of CBTL regeneration from offset curve

3.3 Regenerating CBTLs based on an offset curve

A point cloud must be generated around a concave vertex to lie on a tangent plane outside the region. Around a concave vertex, however, the point cloud is generated inside the region because the cross boundary derivative is generated inside the boundary edge. Due to this, in our method, the direction of the cross boundary derivative is corrected and an offset curve is generated according to the cross boundary derivative, so that a CBTL is generated outside the region. As a result, a point cloud is generated outside the region so as to lie on a reference plane. If the offset value is too large, the cross boundary derivative will self-interfere. To avoid the self-interference, the offset value is controlled when the offset curve is generated.

3.3.1 Cross boundary derivative generation

In our method, a Tangent Ribbon[1] is assumed with the cross boundary derivative. When one of the end points of a boundary edge is a concave vertex, as

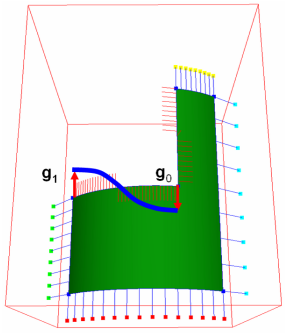


Fig. 9: Correcting the direction of cross boundary derivative

shown in Figure 9, the Tangent Ribbon will be twisted and the cross boundary derivative is introduced in the opposite direction from a certain position. In the case of the shape of Figure 9, cross boundary derivative $\mathbf{g}_0$ on the concave vertex is generated toward the inside of the N-sided region. Then, the cross boundary derivative around the edge is derived after the direction of the vector $\mathbf{g}_0$ becomes opposite. This operation maintains the consistency of the tangent plane at the vertex.

Figure 10 shows two adjacent Bezier surfaces $\mathbf{S}^1$ and $\mathbf{S}^2$. The Bezier surface with control point $\mathbf{P}_{ij}$ ($i = 0,\cdots,3$; $j = 0,\cdots,3$) is shown in equation (6), where $B_i^n(u)$ and $B_j^m(v)$ are Bernstein base polynomials[1].

$$\mathbf{S}(u,v) = \sum_{i=0}^{n}\sum_{j=0}^{m} B_i^n(u) B_j^m(v) \mathbf{P}_{ij} \qquad (6)$$

In order that two adjacent surfaces are G^1-continuous, the cross boundary derivative obtained from the points on the boundary edges need to satisfy equation (7)[13].

$$\mathbf{S}_u^2(0,v) = k(v)\mathbf{S}_u^1(1,v) + h(v)\mathbf{S}_v^1(1,v) \qquad (7)$$

where $k(v)$ and $h(v)$ are the scalar functions of v. $\mathbf{S}_u^2$, $\mathbf{S}_u^1$ and $\mathbf{S}_v^1$ are obtained using equation (8):

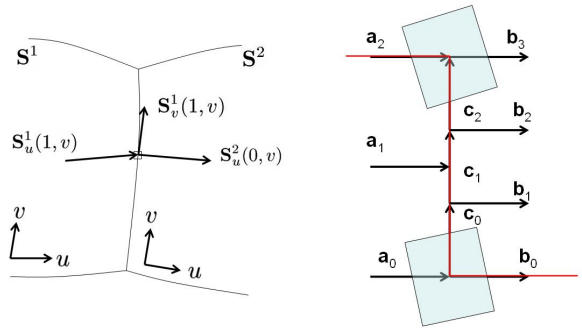


Fig. 10: Vectors at end points

$$\begin{aligned}\mathbf{S}_u^2(0,v) &= \frac{\partial \mathbf{S}^2(0,v)}{\partial u}\\ \mathbf{S}_u^1(1,v) &= \frac{\partial \mathbf{S}^1(1,v)}{\partial u}\\ \mathbf{S}_v^1(1,v) &= \frac{\partial \mathbf{S}^1(1,v)}{\partial v}\end{aligned} \tag{8}$$

When $v = 0$ and $v = 1$ are assigned to equation (7), equation (9) is obtained:

$$\begin{aligned}\mathbf{b}_0 &= k_0\mathbf{a}_0 + h_0\mathbf{c}_0\\ \mathbf{b}_3 &= k_1\mathbf{a}_3 + h_1\mathbf{c}_2\end{aligned} \tag{9}$$

where k_0, k_1, h_0 and h_1 are real numbers, $k_0 > 0$, $k_1 > 0$ and $\mathbf{a}_0$, $\mathbf{a}_3$, $\mathbf{b}_0$, $\mathbf{b}_3$, $\mathbf{c}_0$ and $\mathbf{c}_2$ are the vectors between the control points. In our method, if $h_0 = 0$ and $h_1 = 0$ are assumed, equation (10) is obtained:

$$\begin{aligned}\mathbf{b}_0 &= k_0\mathbf{a}_0\\ \mathbf{b}_3 &= k_1\mathbf{a}_3\end{aligned} \tag{10}$$

Here, to satisfy equation (9), the scalar function $k(v)$ about v is assumed to be a linear function.

$$k(v) = k_0(1-v) + k_1 v \tag{11}$$

From equations (7) and (11), the equation (12) is obtained using the vectors between the control points of a surface (Figure 10):

$$\sum_{i=0}^{3} B_i^3(v)\mathbf{b}_i = \{k_0(1-v) + k_1 v\}\sum_{i=0}^{2} B_i^2(v)\mathbf{a}_i \tag{12}$$

Since the left side of equation (12) becomes cubic, the degree of polynomial $\mathbf{a}_i$ is limited to be quadratic. Therefore, when we assume $\mathbf{a}_i$ using basis patch method[12], the unit vectors $\mathbf{a}_0$ and $\mathbf{a}_2$ are calculated from the boundary and $\mathbf{a}_1$ is calculated using the equation (13):

$$\mathbf{a}_1 = \frac{\mathbf{a}_0 + \mathbf{a}_2}{2} \tag{13}$$

Since the cross boundary derivative is generated in the opposite direction on the concave vertex in Figure 9, $\mathbf{S}_u^2(0,v) = -\mathbf{g_0}$ is obtained. When $v = 0$, equation (14) is obtained:

$$\mathbf{a}_0 = -k_0\mathbf{g}_0 \tag{14}$$

If equation (12) is solved, the following is obtained:

$$\begin{aligned}\mathbf{b}_1 &= -\frac{(k_0+k_1)\mathbf{g}_0}{3} + \frac{k_0\mathbf{g}_1}{3k_1}\\ \mathbf{b}_2 &= -\frac{k_0\mathbf{g}_0}{3} + \frac{(k_0+k_1)\mathbf{g}_1}{3k_1}\end{aligned} \tag{15}$$

From equations (15), the cross boundary derivative that corrects the direction of the point on the boundary edge is obtained. As a result, the cross boundary derivative is generated in the direction of the outside of the region. The cross boundary derivative on the point of the boundary edge is calculated. CBTLs are generated from normalized cross boundary derivative vectors and the same length t is applied to the following equation:

$$\mathbf{P}(v) = \mathbf{Q}(v) + t\frac{\mathbf{S}_u^2(0,v)}{|\mathbf{S}_u^2(0,v)|} \tag{16}$$

where $\mathbf{P}(v)$ is the end point of the CBTL, $\mathbf{Q}(v)$ is the point on the boundary edge at parameter v.

3.3.2 Offset curve generation

If our method is applied to an L-shaped region, such as the one shown in Figure 6 (a), the CBTLs intersect each other. Since the intersection will cause a distorted offset curve, the intersecting CBTLs, enclosed with the red circle in Figure 11, are deleted. After

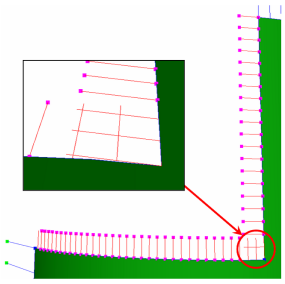


Fig. 11: Deleting intersecting CBTLs

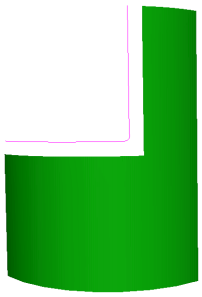


Fig. 12: Generated offset curve

that, the end points of the CBTLs are approximated with a B-spline curve so that an offset curve is generated. Figure 12 shows the generated offset curve.

The CBTLs are regenerated between each point on the boundary edges and the corresponding point on the generated offset curve. Then, every generated CBTL is divided into four segments and the sample points are extracted as the point cloud.

Since the intersecting CBTLs are deleted upon offset curve generation, necessary data may be lacking. To compensate this, around the concave vertex where data is lacking, necessary CBTLs are generated by finding the highest curvature point of the concave vertex from the offset curve.

3.4 Method of generating boundary curves

This section describes the method of generating four boundary curves of the B-spline surface from an N-sided region that involves a concave shape. As is described in section 2.3, the method[10] may generate CBTLs so that they intersect each other or they are

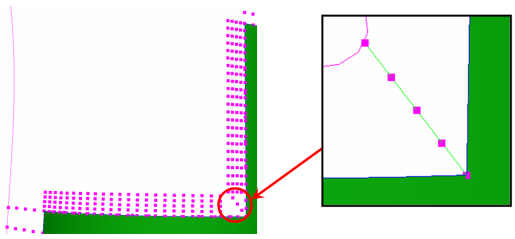


Fig. 13: Regenerating a CBTL through reverse-search

twisted. This causes distortion of boundary curves. To avoid distortion, boundary curves are generated by CBTLs obtained from convex edges.

Take an example of a L-shaped concave region. Figure 14 is a result of obtaining the CBTLs from convex edges and generating the boundary curves of a B-spline surface. Each of the four curves is an independent B-spline curve, so that the end points of the curves must coincide to form a closed region. Then, as shown in Figure 14, since the end points of the boundary curves must coincide, each of the end points of the boundary curves is extended to the adjacent boundary plane in the tangent direction. For instance, when the points on the reference plane are assumed to be P_1 and P_2, these points are averaged to obtain point P_3, which is set as the end point of the curves. We applied our method to 536 surfaces. As a result, we judged that the influence on the surface accuracy caused by this operation was not large. In the same manner, four corner points of the B-spline surface are determined. Each of the boundary curves is re-fitted by adding the corner points. Figure 15 shows the result of generating the closed boundary curves. Since edges connected to the concave vertex or end points of CBTLs generated from the concave edge are excluded, the boundary curves are not distorted.

3.5 Procedure of generating a new surface

This section describes the procedure to apply a surface to a region according to our method. The sign $*$ indicates new processing steps. In our method, boundary edges are input and a bicubic B-spline surface is output. The flowchart of our method is shown in Figure 16.

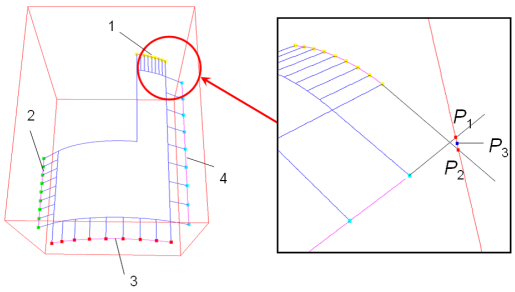


Fig. 14: Four independent boundary curves

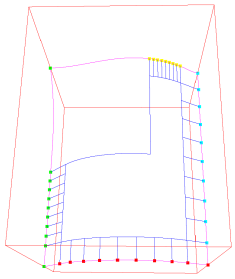


Fig. 15: Example of generating boundary curves for a region with a concave vertex

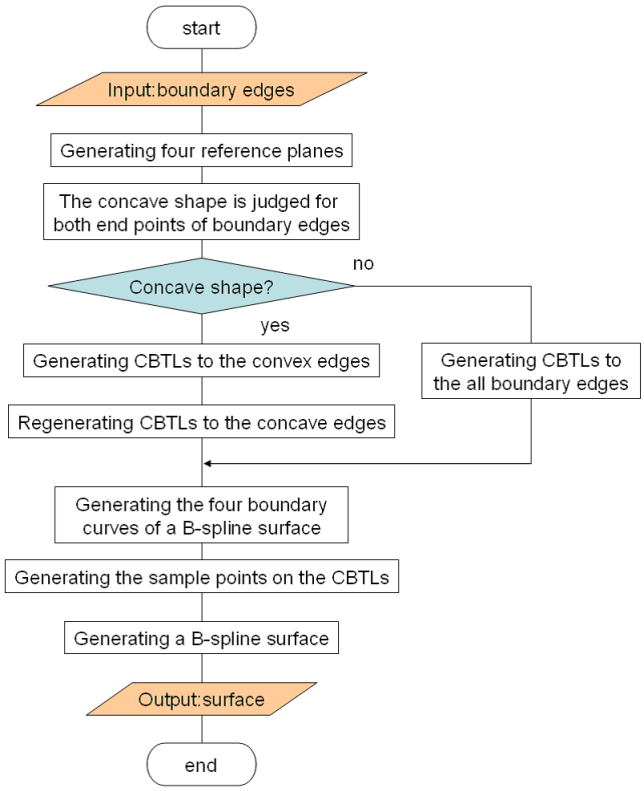


Fig. 16: Flowchart of our method

∗ **Step1:** Using the method described in section 3.1, four reference planes that enclose the boundary edges are extracted.

∗ **Step2:** The concave shape is judged for both end points of the boundary edge, and the edge whose end points are convex vertices is selected. For the selected edge, the processing similar to Step 2 of section 2.2 is performed.

If a boundary edge is a concave one, like the edges that generate CBTLs shown in the red circles in Figure 6 (b), all the CBTLs of the boundary edge are deleted.

If there are one or more intersection points of the CBTLs, as shown in the red circle in Figure 6 (c), all the CBTLs of the corresponding boundary curve are calculated again using the method described in section 3.2.

∗ **Step3:** The offset curve is generated using the edge connected to the concave vertex as described in section 3.3, and then CBTLs are regenerated. Then, a point cloud is generated on the CBTLs. The generated point cloud lies on the tangent plane at the sample points of the boundary edge. A point cloud is generated on the boundary edge representing a hole, and is added to the sample points. In our method, surface fitting is performed only to the point cloud generated on CBTLs. The machine parts have few undulating surfaces. Therefore, we judged that the influence on surface accuracy is not large even if only the point cloud generated on CBTLs is used.

∗ **Step4:** The boundary curves of the B-spline surface is generated as described in section 3.4.

Step5: From the point cloud generated in Step 3 and the boundary curves generated in Step 4, the inner control points of the B-spline surface are calculated using the least-squares method. Details of the approximation method are described in the appendix.

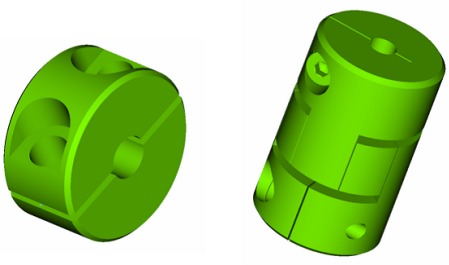


Fig. 17: JIS model used in our experiment

4 Experimental Results

Our method was applied to an N-sided region representing a notch including concave shapes. The generated surface shape was evaluated and the accuracy was verified.

4.1 B-spline surface generation

Our method was applied to the boundaries of a trimmed surface obtained from practical data of the JIS models[2] shown in Figure 17. Figures 18 and 19 show the result of generation of new trimmed surfaces. (a) shows the shaded image of the generated trimmed surface, (b) shows the point cloud extracted from the CBTLs, (c) shows the error evaluation and (d) shows the control points of the generated B-spline surface.

4.2 Method of shape evaluation

To verify the accuracy, the distance between the generated surface and the source surface that the trimmed surface retained was measured. The source surface was divided equally in both u and v directions into twenty sections so that a square grid was generated. The generated grid points lying inside the trimmed surface were extracted. The extracted point cloud was projected to the generated B-spline surface and the shortest distance was measured.

In addition, in order to confirm that the source boundary edges lie on the generated B-spline surface within a tolerance, the distance between the source boundary edge and the generated surface was measured. Figures 18 (c) and 19 (c) show the result of shape evaluation. Figures 18 (c) and 19 (c) show the

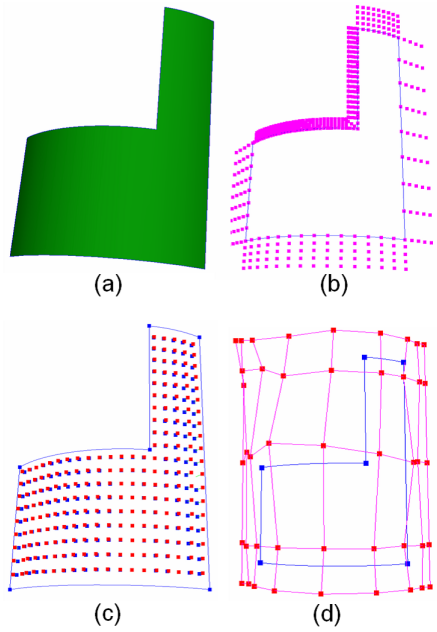


Fig. 18: Shape A with a notch

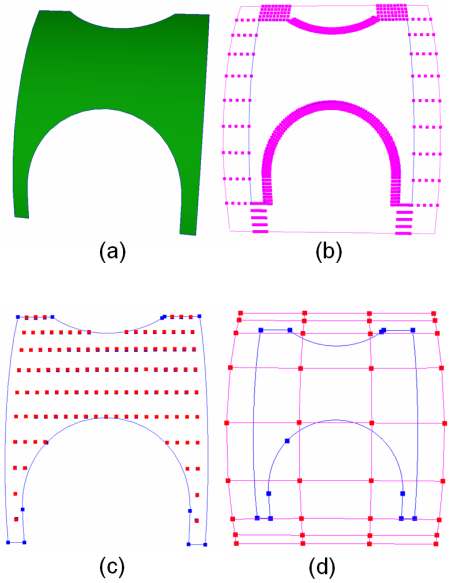


Fig. 19: Shape B with a notch

distance between the source surface and the generated one. The blue dots are put by projecting the grid points of the source surface to the generated surface. The red dots are put by extending the blue dots in the normal direction of the tangent plane. The distance between a blue dot and a red one represents the distance between the source surface and the generated one multiplied by twenty five.

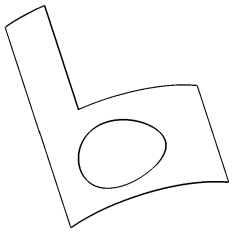


Fig. 20: Concave shape with a hole

4.3 Comparison and shape evaluation of methods

In this section, our method is compared with the method of Garcia et al.[9] and Tokuyama et al.[10]. The existing methods and our method are applied to the same concave shape with a hole shown in Figure 20 and the generated surfaces are compared to show the effectiveness of our method.

Since the method of Garcia et al. does not deal with the shape with holes, the method cannot generate a surface for the shape of Figure 20. Due to this, the method of Tokuyama et al. and our method are applied to the shape of Figure 20 and the generated surfaces are evaluated.

The result of applying the Tokuyama's method is shown in Figure 21 and that of applying our method is shown in Figure 22. In each of Figures 21 and 22, (a) shows the shaded image of the generated surface and (b) shows the parameter line of the generated surface. Applying our method can remove the distortion of the generated surface. Table 1 shows the result of shape evaluation of the generated surface. In Table 1, three kinds of values are shown: " Avg " indicates the average error margin value obtained by averaging the distances between the generated surfaces and the source one, " Max " indicates the maximum error margin value representing the maximum distance between the generated surface and the source one, " Ratio " indicates the ratio of the bounding box size and the maximum distance. In addition, in order to verify whether the boundary edges lie on the generated surface within a tolerance, the average error margin value, the maximum error margin value, and the ratio are also shown in Table 1. From the values, we can find that our method can generate the surface with better accuracy than the existing method.

Table 1: Comparison of the generated surfaces by each method(Method1:Tokuyama et al., Method2:our method, Avg:Average error, Max:Maximum error)

Method	Evaluation object	Avg	Max	Ratio(%)
Method1	Trimmed surface	0.1087	0.8163	3.836
	Boundary edges	0.1137	2.6637	12.51
Method2	Trimmed surface	0.0059	0.0246	0.115
	Boundary edges	0.0018	0.0088	0.041

Table 2 shows the result of shape evaluation performed for the nine surfaces including notches and holes, obtained from the JIS model of Figure 17. Figure 23 shows the shape of the data listed in Table 2. As shown in Table 2, the trimmed surface is approximated with good accuracy.

4.4 Verification of effectiveness on direct modeling

To verify the effectiveness for direct modeling, one of the boundary edges of the shape as shown in Figure 20 was modified as shown in Figure 24 (a) and our method was applied to the modified boundary edge. Figure 24 (b) shows the shaded image of the result. We can find that the surface is generated without distortion for the modified boundary edge. The result of shape evaluation of the generated surface is shown in Table 3. Since the surface shape and the boundary edge are modified separately, the geometrical consistency of them will collapse. Therefore, the error margin of the generated surface and the source surface cannot be evaluated. As shown in Table 3, we can find that the surface is generated so that the boundary edge lies on it within a small error margin.

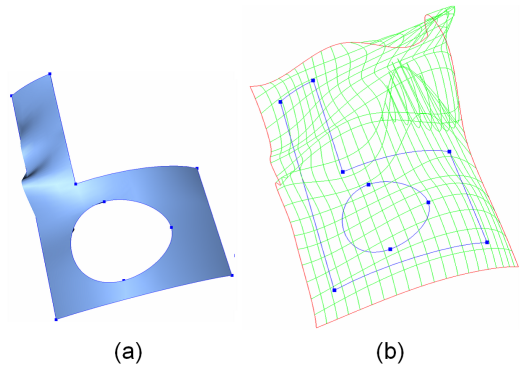


Fig. 21: Result of applying the method of Tokuyama et al.

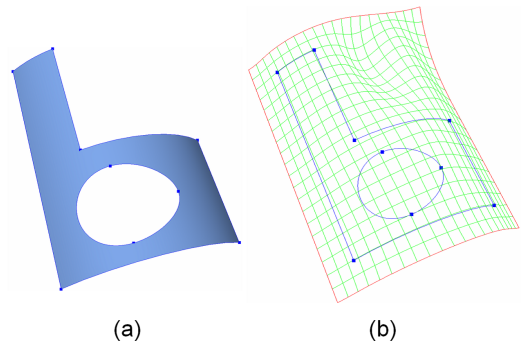


Fig. 22: Result of applying our method

Table 2: Evaluation of errors between the source surface and the generated one

Shape	Evaluation object	Avg	Max	Ratio(%)
a	Trimmed surface	0.0064	0.0206	0.096
	Boundary edges	0.0034	0.0096	0.048
b	Trimmed surface	0.0017	0.0034	0.019
	Boundary edges	0.0020	0.0044	0.024
c	Trimmed surface	0.0032	0.0082	0.039
	Boundary edges	0.0017	0.0037	0.018
d	Trimmed surface	0.0051	0.0286	0.156
	Boundary edges	0.0014	0.0038	0.021
e	Trimmed surface	0.0075	0.0233	0.109
	Boundary edges	0.0032	0.0143	0.067
f	Trimmed surface	0.0028	0.0077	0.037
	Boundary edges	0.0012	0.0033	0.016
g	Trimmed surface	0.0026	0.0064	0.044
	Boundary edges	0.0007	0.0032	0.022
h	Trimmed surface	0.0067	0.0231	0.180
	Boundary edges	0.0014	0.0049	0.041
i	Trimmed surface	0.0060	0.0220	0.100
	Boundary edges	0.0025	0.0129	0.060

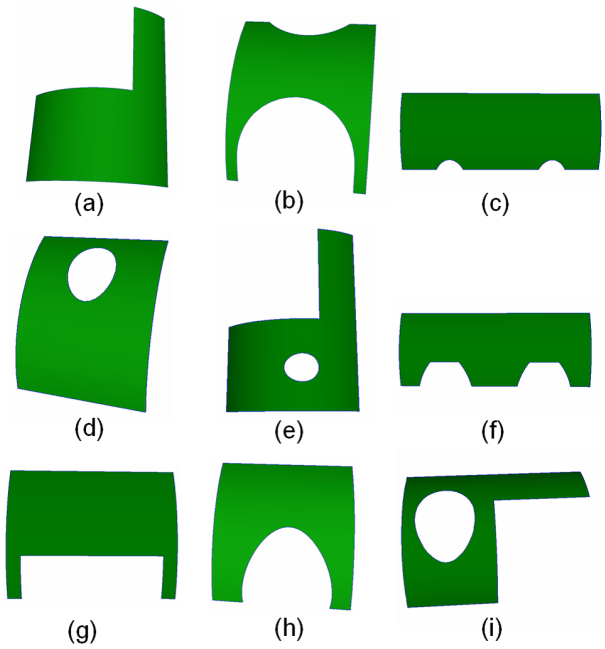


Fig. 23: Surface shapes used in our experiment

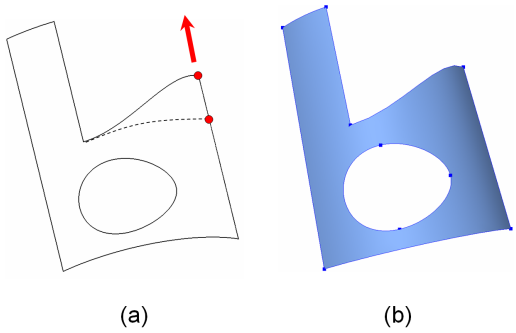


Fig. 24: Result of applying our method to modified boundary edge

Table 3: Evaluation of errors between the generated surface and the modified boundary edge

Evaluation object	Avg	Max	Ratio(%)
Boundary edges	0.0026	0.0072	0.029

5 Conclusion

In this paper, we proposed the method of fitting a B-spline surface to an N-sided region with notches, without distortion. In our method, the boundary curves of a B-spline surface were generated with considering the shape around a concave. The offset curves were also generated to compensate the deleted CBTLs, which supplement the insufficient sample points around the concave shape.

Our method was applied to the surface shape with notches and holes, obtained from a JIS model and the practicality was verified. The distances between the trimmed surface and the B-spline surface generated using our method were measured and an excellent result was obtained. The source surface of the trimmed surface can be estimated from the boundary curves and the effectiveness for direct modeling was confirmed. In our method, continuity with the adjacent surfaces is not maintained. Our future work is to extend this method so as to consider the continuity with the adjacent surfaces. The basic concept of our method has already been presented in NICOGRAPH INTERNATIONAL 2010[15]. This paper describes the details of the implementation.

Acknowledgments

We would like to thank Yamamoto for the experiment. A part of the study is supported by the JST seeds excavation examination in 2009 fiscal year.

References

[1] G. Farin: Curves and Surfaces for Computer Aided Geometric Design A Practical Guide, Academic Press, (1996)

[2] web2CAD, http://www.web2cad.co.jp/

[3] Y. Muraki, K. Konno, Y. Tokuyama: “A STUDY OF SUBDIVISION METHOD TO THREE AND FIVE SIDED FACES BASED ON REGULAR POLYGON”, Proc. of IWAIT 2009, (2009)

[4] L. Piegl, W. Tiller: “Filling n-sided regions with NURBS patches”, The Visual Computer, Vol.2, No.15, pp.77-89, (1999)

[5] Yi-Jun. Yang, Jun-Hai. Yong, Hui. Zhang, Jean-Claude. Paul, Jia-Guang. Sun: “A rational extension of Piegl’s method for filling n-sided holes”, Computer Aided Design, Vol.38, No.11, pp.1166-1178, (2006)

[6] T. W. Sederberg, D. L. Cardon, G. T. Finnigan, N. S. North, J. Zheng, and T. Lyche: “T-spline Simplification and Local Refinement”, SIGGRAPH 2004, Vol.23, No.3, pp.276-283, (2004)

[7] E. Catmull, J. Clark “Recursively generated B-spline surfaces on arbitrary topological meshes”, Computer Aided Design, Vol.10, No.6, pp.350-355, (1978)

[8] C. Deng, X. Yang “A simple method for interpolating meshes of arbitrary topology by Catmull-Clark surfaces”, The Visual Computer, Vol.26, No.2, pp.137-146, (2009)

[9] N. Pla-Garcia, M. Vigo-Anglada, J. Cotrina-Navau: “N-sided patches with B-spline boundaries”, Computers & Graphics, Vol.30, No.6, pp.959-970, (2006)

[10] Y. Tokuyama, K. Konno: “Filling N-sided Region with a B-spline Surface”, Information Processing Society of Japan, Vol.43, No.10, pp.3209-3218, (2002)

[11] L. Piegl, W. Tiller: The NURBS Book, Springer-Verlag, (1995)

[12] K. Konno, Y. Tokuyama, H. Chiyokura: “A G^1 connection around complicated curve meshes using C^1 NURBS Boundary Gregory Patches”, Computer Aided Design, Vol.33, No.4, pp.293-306, (2001)

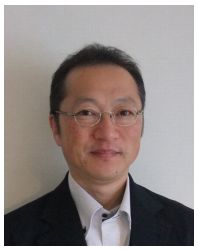
[13] H. Toriya, H. Chiyokura: Basics and Applications of Three-dimensional CAD, Kyoritsu Shuppan Co., Ltd., (1991)

[14] K. Ueda: “Mean normal vector to a surface bounded by Bezier curves”, Computer Aided Geometric Design, Vol.13, No.5, pp.441-451, (1996)

[15] Y. Muraki, K. Konno, Y. Tokuyama: “Curve Mesh Modeling Method of Trimmed Surfaces for Direct Modeling”, Proc. of NICOGRAPH INTERNATIONAL 2010, (2010)

Authors' biographies:

Yuta MURAKI is a Post-doctoral Fellow in Faculty of Engineering at Iwate University. He received the B.E. and M.E. degrees in Computer and Information Sciences, and the D.E. degree in Electronic Information Science from Iwate University in 2005, 2007, and 2010, respectively. His research interests include geometric modeling, CG, CAD and 3D measurement systems. He is a member of JSPE.



Kouichi KONNO is a professor of Faculty of Engineering at Iwate University. He received a BS in Information Science in 1985 from the University of Tsukuba. He earned his Dr.Eng. in precision machinery engineering from the University of Tokyo in 1996. He joined the solid modeling project at RICOH from 1985 to 1999, and the XVL project at Lattice Technology in 2000. He worked on an associate professor of Faculty of Engineering at Iwate University from 2001 to 2009. His research interests include virtual reality, geometric modeling, 3D measurement systems, and computer graphics. He is a member of IEEE CS.

Yoshimasa TOKUYAMA is a professor of Department of Media and Image Technology, Faculty of Engineering, Tokyo Polytechnic University. He received his MS in Mechanical Engineering in 1986 and doctor degree in Computer Graphics in 2000 from The University of Tokyo. He was a member of the 3D CAD project at RICOH 's Software Division from 1986 to 2002. His areas of research interest include computer graphics, game, haptic interface, virtual reality. He is a member of Information Processing Society of Japan, Institute of Image Information and Television Engineers of Japan, the Institute of Image Electronics Engineers of Japan, the Society for Art and Science.

Appendix

Approximation method by using the least-squares method

This section describes the outline of the approximation method by using the least-squares method[11]. In our method, a surface is generated with the following procedures by using the sample points generated in Step 3 and the boundary curve generated in Step4 of section 3.5.

(a) Presume the parameter values of the sample points other than the point cloud, used upon boundary curve generation.

(b) Calculate the control points of a surface by using the least-squares method.

(c) Correct the parameter values of each point on the surface and return to (b).

(d) Evaluate the distance between the surface and the point. If all the values are smaller than the specified error margin, finish the processing.

Presumption of u and v parameter values of the point cloud

The point cloud approximation requires u and v parameters of a given point cloud. The surface on which sample point $\mathbf{Q}$ lies is assumed to be $\mathbf{S}(u, v)$. Since surface $\mathbf{S}$ is unknown in (a), a ruled surface is assumed first as initial surface $\mathbf{S}$ and sample point $\mathbf{Q}$ is projected onto the surface. The u and v parameters of the projected point $\mathbf{Q}'$ are set to u and v parameter values of sample point $\mathbf{Q}$. In (c), the parameter values are calculated by using sample point $\mathbf{Q}$ and surface $\mathbf{S}$ generated in (b).

Surface control point calculation by least-squares method

In (b), unknown control points are calculated by using the least-squares method. When the fitting B-spline surface is $\mathbf{S}(u, v)$, the surface control points are $\mathbf{P}_{i,j}(0 \leq i \leq n, 0 \leq j \leq m)$. Surface $\mathbf{S}(u, v)$ is expressed by equation (17).

$$\mathbf{S}(u,v) = \sum_{i=0}^{n}\sum_{j=0}^{m} N_{i,k}(u) M_{j,k}(v) \mathbf{P}_{i,j} \tag{17}$$

where $k = 3$. The $t+1$ sample points are assumed to be $\mathbf{Q}_s(0 \leq s \leq t)$ and the parameters of u and v directions are assumed to be $\overline{u}_s$ and $\overline{v}_s$. Surface $\mathbf{S}(u, v)$ is calculated so that the square sum of the distances between the sample points $\mathbf{Q}_s$ and corresponding points $\mathbf{S}(\overline{u}_s, \overline{v}_s)$ on the surface is minimized. Then, equation (18) is obtained.

$$f = \sum_{s=0}^{t} |\mathbf{Q}_s - \mathbf{S}(\overline{u}_s, \overline{v}_s)|^2 \tag{18}$$

In equation (17), the control point showing the boundary curve is already known. When the known control point is assumed to be $\mathbf{R}_s$, equation (20) is obtained.

$$\begin{aligned}
\mathbf{R}_s =& \mathbf{Q}_s - \big(N_{0,k}(\overline{u}_s) M_{0,k}(\overline{v}_s) \mathbf{P}_{0,0} + \cdots \\
&+ N_{0,k}(\overline{u}_s) M_{m,k}(\overline{v}_s) \mathbf{P}_{0,m} \\
&+ N_{1,k}(\overline{u}_s) M_{0,k}(\overline{v}_s) \mathbf{P}_{1,0} + \cdots \\
&+ N_{n-1,k}(\overline{u}_s) M_{0,k}(\overline{v}_s) \mathbf{P}_{n-1,0} \\
&+ N_{1,k}(\overline{u}_s) M_{m,k}(\overline{v}_s) \mathbf{P}_{1,m} + \cdots \\
&+ N_{n-1,k}(\overline{u}_s) M_{m,k}(\overline{v}_s) \mathbf{P}_{n-1,m} \\
&+ N_{n,k}(\overline{u}_s) M_{0,k}(\overline{v}_s) \mathbf{P}_{n,0} + \cdots \\
&+ N_{n,k}(\overline{u}_s) M_{m,k}(\overline{v}_s) \mathbf{P}_{n,m}\big)
\end{aligned} \tag{19}$$

$$\begin{aligned}
f &= \sum_{s=0}^{t} |\mathbf{Q}_s - \mathbf{S}(\overline{u}_s, \overline{v}_s)|^2 \\
&= \sum_{s=0}^{t} \Bigg[\mathbf{R}_s \cdot \mathbf{R}_s - \mathbf{2} \sum_{\mathbf{i=1}}^{\mathbf{n-1}} \sum_{\mathbf{j=1}}^{\mathbf{m-1}} \mathbf{N_{i,k}}(\overline{\mathbf{u}}_\mathbf{s}) \mathbf{M_{j,k}}(\overline{\mathbf{v}}_\mathbf{s}) (\mathbf{R_s} \cdot \mathbf{P_{i,j}}) \\
&\quad + \left(\sum_{i=1}^{n-1} \sum_{j=1}^{m-1} N_{i,k}(\overline{u}_s) M_{j,k}(\overline{v}_s) \mathbf{P}_{i,j} \right) \cdot \\
&\qquad \left(\sum_{i=1}^{n-1} \sum_{j=1}^{m-1} N_{i,k}(\overline{u}_s) M_{j,k}(\overline{v}_s) \mathbf{P}_{i,j} \right) \Bigg]
\end{aligned} \tag{20}$$

If equation (20) is differentiated by unknown control point $\mathbf{P}_{\alpha,\beta}$ and the differentiation value becomes 0, the value of f is minimized.

$$\frac{\partial f}{\partial \mathbf{P}_{\alpha,\beta}} = 0 \tag{21}$$

where $1 \leq \alpha \leq n-1$ and $1 \leq \beta \leq m-1$. From equations (20) and (21), equation (22) is obtained.

$$\sum_{i=1}^{n-1}\sum_{j=1}^{m-1}\left(\sum_{s=0}^{t} N_{\alpha,k}(\overline{u}_s)M_{\beta,k}(\overline{v}_s)N_{i,k}(\overline{u}_s)M_{j,k}(\overline{v}_s)\right)\mathbf{P}_{i,j} = \sum_{s=0}^{t} N_{\alpha,k}(\overline{u}_s)M_{\beta,k}(\overline{v}_s)\mathbf{R}_s \tag{22}$$

When $(n-1)\times(m-1)$ equations are set up for $\mathbf{P}_{\alpha,\beta}$ $(1 \leq \alpha \leq n-1, 1 \leq \beta \leq m-1)$ of equation (22), the following matrix is obtained.

$$\mathbf{NP} = \mathbf{R} \tag{23}$$

where

$$\mathbf{N} = \begin{bmatrix} \mathbf{A}_{1,1,i,j} \\ \mathbf{A}_{2,1,i,j} \\ \vdots \\ \mathbf{A}_{\alpha,\beta,i,j} \\ \vdots \\ \mathbf{A}_{n-1,m-1,i,j} \end{bmatrix} \tag{24}$$

$$\mathbf{A}_{1,1,i,j} = \left[\sum_{s=0}^{t} N_{1,k}(\overline{u}_s)M_{1,k}(\overline{v}_s)N_{1,k}(\overline{u}_s)M_{1,k}(\overline{v}_s), \cdots, \sum_{s=0}^{t} N_{1,k}(\overline{u}_s)M_{1,k}(\overline{v}_s)N_{i,k}(\overline{u}_s)M_{j,k}(\overline{v}_s), \cdots, \sum_{s=0}^{t} N_{1,k}(\overline{u}_s)M_{1,k}(\overline{v}_s)N_{n-1,k}(\overline{u}_s)M_{m-1,k}(\overline{v}_s)\right] \tag{25}$$

$$\mathbf{A}_{\alpha,\beta,i,j} = \left[\sum_{s=0}^{t} N_{\alpha,k}(\overline{u}_s)M_{\beta,k}(\overline{v}_s)N_{1,k}(\overline{u}_s)M_{1,k}(\overline{v}_s), \cdots, \sum_{s=0}^{t} N_{\alpha,k}(\overline{u}_s)M_{\beta,k}(\overline{v}_s)N_{i,k}(\overline{u}_s)M_{j,k}(\overline{v}_s), \cdots, \sum_{s=0}^{t} N_{\alpha,k}(\overline{u}_s)M_{\beta,k}(\overline{v}_s)N_{n-1,k}(\overline{u}_s)M_{m-1,k}(\overline{v}_s)\right] \tag{26}$$

$$\mathbf{R} = \begin{bmatrix} N_{1,k}(\overline{u}_0)M_{1,k}(\overline{v}_0)\mathbf{R}_0 + \cdots + N_{1,k}(\overline{u}_t)M_{1,k}(\overline{v}_t)\mathbf{R}_t \\ \vdots \\ N_{n-1,k}(\overline{u}_0)M_{m-1,k}(\overline{v}_0)\mathbf{R}_0 + \cdots + N_{n-1,k}(\overline{u}_t)M_{m-1,k}(\overline{v}_t)\mathbf{R}_t \end{bmatrix} \tag{27}$$

$$\mathbf{P} = \begin{bmatrix} \mathbf{P}_{1,1} \\ \vdots \\ \mathbf{P}_{n-1,m-1} \end{bmatrix} \tag{28}$$

Solving equation (23) obtains unknown control points $\mathbf{P}_{i,j}$.