

Implemented Algorithms of Challenging CAD Technology aimed at Increasing Efficiency of Class A Surfacing

Tomoyoshi SHINOHARA*

Keiko SUNOHARA*

Saburo SAKAI*

Hirotsugu KATSUBE*

Daisuke IDE*

ABSTRACT

In order to satisfy a development target for a 50% reduction in the man hours expended in the generation of Class A surfaces, the development of the system discussed in this paper focused on three phases of modeling leading to the completion of Class A surfaces. New technologies were developed to innovatively increase work efficiency in each area, and these were realized as CAD functions. This paper will discuss the technologies that helped to realize the functions that have had the greatest effect on performance.

First, for the generation of basic surfaces fitted over scanned clay model data, a technology was developed that defines surface behavior and quality as an optimization model based on hints from the stiffness of physical plates. The solution of the optimization model generates high-quality fitting surfaces.

Next, the computational algorithms for the generation of the fillet surfaces that represent approximately 10% of the number of Class A surfaces were fundamentally revised, resulting in a simultaneous increase in speed and accuracy.

For the final phase of Class A data generation, a target-driven modeling (TDM) function was realized that helps enable designers to make minor adjustments in accordance with their aesthetic judgment. This function has increased the efficiency of finishing work for Class A surfaces, which demands the expression of finely detailed design shapes.

1. Introduction

In order to reduce the man-hours necessary for the creation of Class A surface data by 50%, the project discussed in this paper focused on three phases of the modeling process up to the stage of completion of the Class A surfaces, and sought to develop CAD functions which would increase work efficiency in each phase⁽¹⁾. (A) to (C) below were established as the functional conditions necessary to increasing efficiency in the three work processes:

- (A) In the phase of creating fitting surfaces from scanned clay model data, the ability to efficiently create surfaces that satisfy Class A data quality standards while accurately reflecting the scanned data.
- (B) In the phase of creating surfaces that express complex styling shapes such as corners and aggregates, the ability to create high-quality surfaces with a high level of system responsiveness.

- (C) The achievement of greater efficiency in time-consuming work processes when attempting to ensure quality based on clear shape targets, for example when styling orientations are adjusted or a design is changed after Class A surfaces have been created.

In the absence of a technology that satisfied Honda's demands in terms of surface quality and computational performance, the development of technologies that would satisfy the functional conditions described above was necessary for achieving the targets set for the new system.

CAD functions satisfying the functional conditions outlined above were realized in the new system through a process of effective technological development. The new technologies employed in the system are described in (1) to (3) below. This paper will provide an introduction to these technologies, which helped to enable the realization of CAD functions.

- (1) Polygon computation was increased in speed in order to help enable interactive manipulation and referencing of

* HG Administration Division

high-volume scanned data. In addition, a technology was developed that helps to enable the creation of surfaces to be fitted on the scanned data based on hints provided by the stiffness of physical plates. This technology creates high-quality fitting surfaces by solving optimization model that define the behavior and quality of the surfaces.

- (2) The expertise incorporated in the Class A surfacing functions of the previous system were carried over and increased in speed and accuracy, while a command set extending to 185 functions was also developed. In particular, the calculation algorithms for fillet surfaces, which represent approximately 10% of the total number of Class A surfaces, were thoroughly revised to increase both speed and accuracy, helping to enable an increase in the speed of calculation for complex fillet shapes of 90% or more against the previous system.
- (3) For correction of Class A surface shapes, the time-consuming work process until completion of the shapes even if the shape targets are clear was increased in efficiency. Target-driven modeling (TDM), a new concept for CAD operations, was introduced and technologies were developed to realize the concept. The realization of TDM as a CAD function has simplified the modeling work involved in the process of correction of shapes.

2. Overview of Polygon Kernel Functions

Pre-processing and shape analysis of the polygon mesh is necessary in the process of creating Class A surface data from polygon mesh scanned data. Because it was necessary to manipulate data interactively despite the fact that the new system works with high-volume data containing several million vertices, it was necessary to create algorithms that would maximize the characteristics of the polygon kernel data structure and memory management mechanism⁽¹⁾. The main functional requirements and the issues to be solved to achieve them were as follows:

- (1) It was necessary to formulate sectional linear shapes in the mesh as polyline sections in order to help enable the model shapes formed from the scanned data to be checked visually, but the achievement of increased efficiency both in the calculation of points of intersection with cutting planes and in the creation of polylines for groups of points of intersection represented an issue.
- (2) It was necessary to project 3D curves on the polygon mesh in order to help enable selection of specific mesh regions and region-setting for surface generation. However, this represented a challenge when there were holes in the polygon mesh or multiple projected curves.
- (3) A function for filling holes in the polygon mesh due to defects in the scanned data or some other causes was necessary, but the high-speed generation of isotropic hole-filling patches irrespective of the shape or the size of the hole represented an issue.
- (4) It was necessary to incorporate a function that would

calculate the degree of divergence between the polygon mesh and the Class A surfaces in order to help enable users to identify differences between the shapes of the Class A surfaces and the scanned data. High-speed calculation of intersections between the lines and the polygon mesh would be needed for calculation of the degree of divergence.

This chapter will discuss the concrete methods employed to resolve issues (1) through (4) in order to help enable the desired polygon kernel functions to be realized.

2.1. Method of Generation of Polyline Sections

For this function, a two-stage processing method was developed in order to resolve technical issues to help enable efficient treatment of interconnection between contiguous points of intersection, as follows:

- (1) Determination of intersection between sectional plane and polygon mesh topological elements (edges and vertices); and
- (2) Search for contiguous points of intersection and connection processing.

For determining intersection, first the position relationships between the sectional plane and all the vertices are determined, following which intersection is judged for the sectional planes and all the edges. When the vertices are on the obverse side of the sectional planes, they are allocated a positive sign, and when they are on the reverse side they are allocated a negative sign. If the distance between a vertex and a sectional plane is within tolerance values, it is defined as an intersecting vertex. Next, intersecting edges are judged. Edges with vertices that have been allocated different signs are defined as intersecting with sectional planes, and the positions of the points of intersection on the edges are calculated. In Fig. 1, the dotted line represents a sectional plane, the black and white circles respectively represent the vertices on the obverse and reverse sides of the sectional plane, and the red circles represent the vertex intersecting with the sectional plane and the points of intersection on the edge.

During connection processing, an arbitrary intersecting edge or intersecting vertex is defined as the starting point,

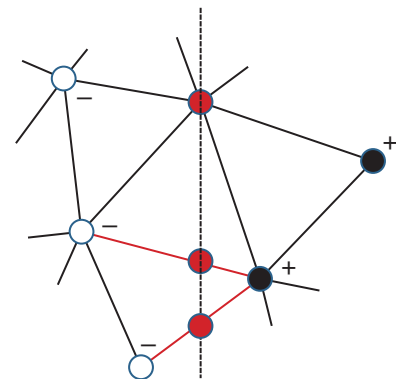


Fig. 1 Polyline-section algorithm

and contiguous intersecting topological elements are successively searched for. These are formed into a single array of points of intersection. In the case of intersecting edges, a search for contiguous intersecting topological elements is commenced from among all the topological elements classified with the faces on both sides, and in the case of intersecting vertices, a search is commenced from among all the topological elements classified with the faces contiguous to the intersecting vertices. One polyline section is completed when all the contiguous topological elements have been determined. The remaining intersecting topological elements are restructured in the same way, and the generation of polyline sections is completed.

2.2. Method of Projecting Curves

It is necessary to correctly output projection calculations even in cases in which the polylines that represent the results of the projections cross holes or the solution consists of multiple polylines. The method used in the new system, which employs the polyline sections discussed in the preceding section, is composed of the following five processing steps. Figure 2 shows an overview of the processing method.

- (1) The curve to be projected is divided into a tessellated polyline.
- (2) The sectional plane is defined from a segment of the tessellated polyline and the direction of projection.
- (3) The polyline of intersection between the sectional plane defined in step (2) and the polygon mesh is determined.
- (4) The projected part of the polyline of intersection that corresponds to the segment is extracted and defined as projected segment.
- (5) A search is conducted for contiguous section fragments by referencing the half-edge data structure; when all the projected segments found in step (4) have been interconnected, the isolation of the curve shape is complete.

2.3. Automatic Hole-filling Processing Method

It was vital to use an algorithm for hole-filling in order

to help ensure polygon quality in the hole-filling patches employed in hole profile shapes. However, performance declined in the case of large hole shapes. The following processing steps were adopted in order to resolve this issue.

- (1) Half-edges with no contralateral elements are detected, and the hole profile is obtained by clustering processing. Figure 3(a) shows the stage of extraction of the hole profile.
- (2) When the hole profile has less than 30 vertices, the Liepa hole-filling algorithm⁽²⁾ is run with conditions set to minimize the total area of the patch, and a hole-filling patch is obtained. Meanwhile, when the hole profile has 30 vertices or more, remeshing is conducted after an initial patch has been generated using the newly developed algorithms in steps (3) to (8).
- (3) The coordinates of the center of the hole profile are determined.
- (4) From among the vertices of the hole profile, new vertices are generated at the point close to the center by the average edge length of the profile polyline, and new edges are extended. At this time, when the distance between contiguous new vertices is lower than a threshold value, the vertices are fused into a single vertex.
- (5) Each contiguous half-edge is tessellated into either a strip shape or a fan shape, generating a band-shaped patch to fill the outer periphery of the hole.
- (6) Steps (1) to (5) are conducted until the small hole in the interior of this belt-shaped patch is smaller than 10 vertices. Figure 3(b) shows a second-generation belt-shaped patch.
- (7) When the size of the hole profile becomes less than 10 vertices, the initial patch is completed by filling the hole in the fan shape with an arbitrary vertex as the starting point. Figure 3(c) shows this stage.
- (8) Remesh processing is conducted for the initial hole-filling patch. In order to obtain an equivalent level of polygon accuracy for the hole-filling patch as for the surrounding shapes, in the remeshing operation, topological rearrangement is repeatedly applied so that the isotropic polygons, the target edge length of which corresponds to

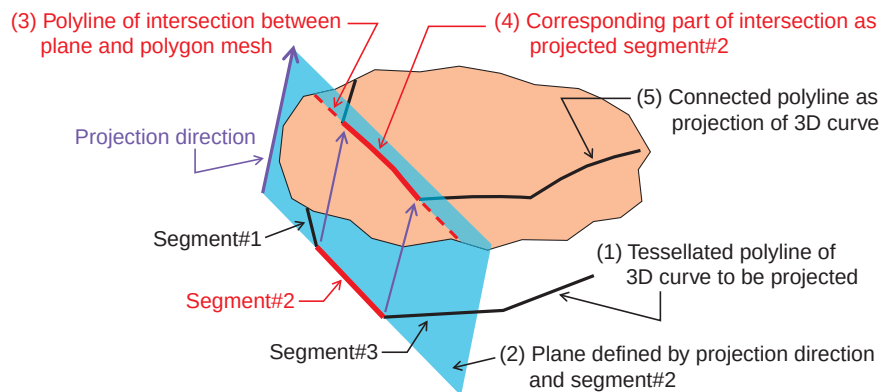


Fig. 2 Algorithm for 3D curve projection on polygon mesh

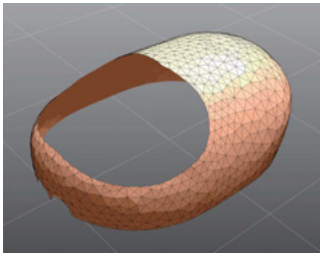


Fig. 3 (a) Hole detection

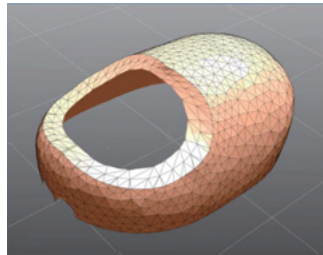


Fig. 3 (b) Initial patch creation

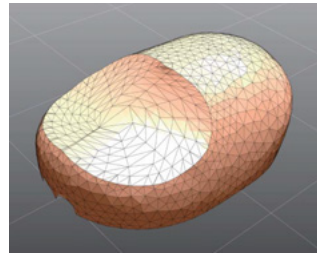


Fig. 3 (c) Completed initial patch

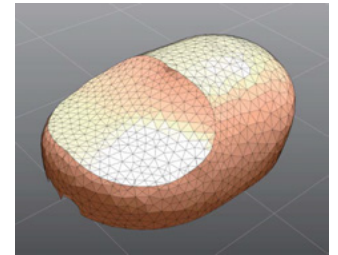


Fig. 3 (d) Hole filled with final patch

the average edge length of the hole profile, become dominant⁽³⁾. The initial patch created in steps (1) to (7) is formed from the initial stage using a set of polygons for which the target edge length has been taken into consideration. This has been of benefit in increasing the speed of remeshing. Figure 3(d) shows the result of completion of hole-filling processing.

2.4. Function for Measurement of Distance between Mesh and Geometric Elements

In order to increase the speed of calculation of intersections between rays and meshes, it was necessary to efficiently create space divisions. This function was realized by means of the following processing steps.

- (1) High-efficiency, high-speed space division
- (2) High-speed intersection calculation using the space divisions created in step (1)
 - Space divisions are created using the following processing steps.
 - (i) Taking an arbitrary polygon as the seed, a cubic region is defined within a fixed distance ($1/40^{\text{th}}$ of the longitudinal diameter of the target mesh) from the center of gravity to each axis.
 - (ii) All the polygons in this region, including the seed, are defined as a component polygon set that forms a grid.
 - (iii) An axis-aligned bounding box (AABB) is calculated for the component polygon set, and this is defined as a grid shape.

- (iv) Targeting polygons that are not classified as part of any grid, steps (i) to (iii) are repeated until all the polygons are classified into either of the grids.

Each grid has overlaps in terms of spatial arrangement, but there is no duplication of the polygon sets incorporated in the grids. That is, any particular polygon will only be assigned to one grid.

Intersection calculations are conducted in two stages. First, intersections with the grids are calculated, following which intersection is calculated for all the polygons inside the grids for which intersection has been determined.

Because all of the grids are created as AABB, judgment of intersections between rays and grids can be conducted at high speed. Figure 4(a) shows a grid aggregate for which intersections with a ray aggregate have been determined in intersection calculations for the target mesh and 10 rays from the exterior. The polygon aggregate inside the grid is shown as pink.

The fact that only this polygon set shown as pink is the focus for judgment of intersections between the rays and the polygons has contributed to reducing the number of necessary calculations, thus boosting performance. The following steps are used to judge intersections between rays and polygons.

- (i) Coordinate transformation is applied to the entire set of vertices of the mesh that is being processed in such a way that the ray and the X axis overlap each other.
- (ii) The polygons are excluded as candidates for intersection

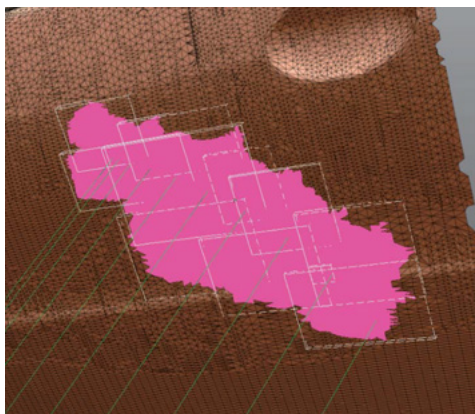


Fig. 4 (a) Ray-grid intersection

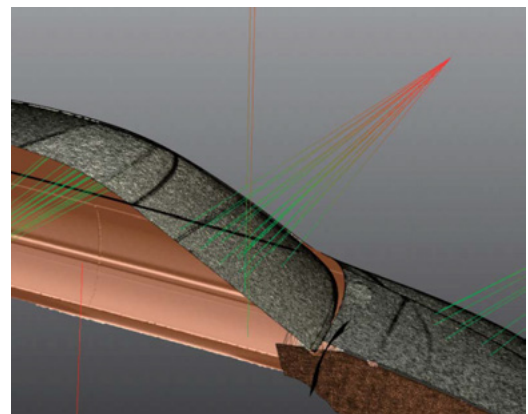


Fig. 4 (b) Ray-mesh intersection

if the signs of all the Y coordinates for three vertices are the same, or if the signs of all the Z coordinates for three vertices are the same.

- (iii) From among the polygons that pass the screening in (ii), all the polygons which intersect with the X axis are defined as intersecting polygons. Since the intersection determination has already been conducted, the calculation of the 3D coordinates of the points of intersection is a simple matter.

Figure 4(b) shows the results of judgment of intersections with multiple rays for the generation of the target mesh.

3. Fitting Technology

Class A data is generated with 3D scanned data from clay models as the input. Formerly, during this process the deviation or styling tendency between the Class A data and the scanned data was judged, and a process of manual correction of the surfaces was repeated. Because this process necessitated an enormous expenditure of man-hours, numerous systems for the automatic generation of surface shapes that reflect styling shapes from scanned data have been considered. This will be defined as “scanned data fitting technology.”

Because undulations in clay models and measurement errors were directly reflected in the output, the previous fitting technology was unable to ensure the necessary surface quality for Class A data. For the new system, the following technologies were therefore developed to remove clay model undulations and measurement errors while maintaining the styling shapes, thus generating surfaces that fulfill Class A data quality demands.

- (1) An intelligent fitting technology that helps to ensure the quality of individual surfaces
- (2) A merge technology that helps to ensure continuity between multiple surfaces

This chapter will discuss the methods employed to realize these technologies.

3.1. Intelligent Fitting Technology

In order to maintain both surface quality and continuity

between surfaces, it is necessary to control the tangency (G1) and curvature (G2) of the surfaces. In addition, as indicated above, undulations in the curvature affect quality, and because of this it is also necessary to control the rate of change of G2, termed G3.

In order to control these parameters, the concept of stiffness has been introduced in the new system. Stiffness here refers to the situation, for example, when a metal ruler is pressed over an actual model: the ruler follows the tendency of the surface, but the minute concavities and convexities are ignored. This is because the ruler has a specific degree of hardness, and will not bend more than a specific amount.

Figure 5 shows an image of the behavior of stiffness. The flesh-colored scanned data incorporates minute differences in level. Commencing with an arbitrary initial surface (shown as blue), surfaces are fitted to minimize the deviation from the scanned data, producing the surfaces shown as purple, which incorporate undulations. In the new system, constraint conditions (shown as the red arrows) are incorporated as the scope of variation in curvature (G3) within the range within which fitting is able to operate. This helps to enable the generation of an intelligent fitting surface (shown as red) which follows the scanned data while undulations are controlled.

Based on this idea, in addition to an approach using the least squares method, the new system employs conditional convergence calculations with the properties of the desired shapes as the conditions. The following objective function and constraint condition are defined for the convergence calculations.

Objective function: $(\text{Deviation from scanned data})^2 + (\text{Value of G3 at each point})$

Constraint condition: $(\text{Value of G3}) < (\text{Upper limit of stiffness})$

The value of G3 was used as a constraint condition because it was considered that it could function to control undulations in the same way as the stiffness of the ruler. An excessive G3 constraint would reduce the ability to trace scanned data, while an insufficient G3 constraint would be prone to produce undulations. Considering this as the hardness of a metal plate, the level of adjustment

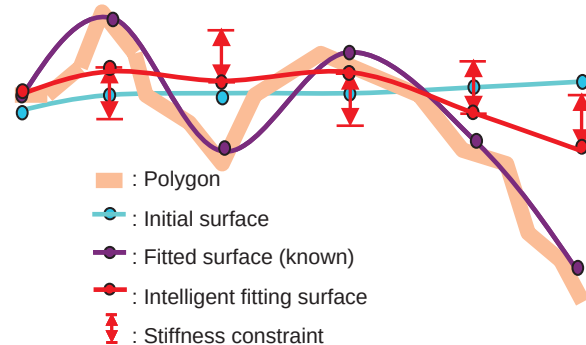
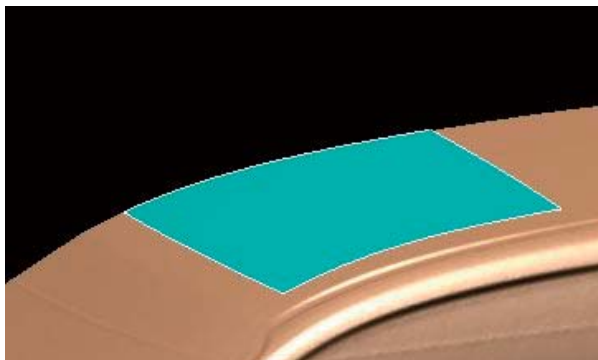


Fig. 5 Stiffness as constraint

of stiffness was applied as a constraint condition, which can be interpreted as altering the thickness of the metal plate. In addition, because undulations are promoted across the entire surface if the value of G3 is increased, this was incorporated in the objective function as an energy function for the convergence calculations.

The concrete methods of incorporating these considerations in the system will be described below.

The input and output of the system can be summarized as follows:

Inputs: 3D scanned data

Four boundary lines defining the area of a surface

Number of control points (CP) for NURBS surfaces

Magnitude of stiffness

Output: Surfaces satisfying shape quality demands

Based on the input data, an initial surface that serves as a standard is generated and fitted. An NURBS surface is used for the mathematical expression of the surface. Within the u and v spaces that can be defined from the input boundary lines, control points are spaced at regular intervals and an initial surface is defined. On the NURBS surface, the surface is expressed by multiplying the basis function $B(u, v)$ by the coordinate values of the control points (CP). The basis function is determined uniquely using the parameter space u, v and the junctures in the space (the knots on the NURBS surface). Now, the equation for the surface, $S(u, v)$ in the parameter space u, v can be expressed as follows using the n control points $a_0, a_1, \dots, a_n$:

$$S(u, v) = (B_0(u, v), B_1(u, v), \dots, B_n(u, v)) \begin{pmatrix} a_0 \\ a_1 \\ \vdots \\ a_n \end{pmatrix} = [B][a] \quad (1)$$

Here, the matrix of the basis function is given as $[B]$.

As Eq. (1) shows, the surfaces can be expressed as linear calculations. Because the number of regions in which surfaces are being fitted and the number of control points are known in advance, the unknown quantity to be determined for the generation of intelligent fitting surfaces is the control point sequence $a_0, a_1, \dots, a_n$.

Because $[B_0(u, v), B_1(u, v), \dots, B_n(u, v)]$ is given, finding the differential of the direction of u, v can be defined as an independent operation to $a_0, a_1, \dots, a_n$. This is defined as the n th-stage differentiation matrix in the u direction, $[D_u^n]$, and used in the calculation of G2 and G3 in Eqs. (9) and (19) below.

$$\partial S / \partial u = [D_u^n][B][a], \quad \partial^2 S / \partial u^2 = [D_u^n][D_u^n][B][a], \quad \partial^3 S / \partial u^3 = [D_u^n][D_u^n][D_u^n][B][a] \quad (2)$$

First, the objective function for the deviation from the measurement points, C_{Dist} is determined. The closest parameter point on the benchmark surface, $(u_0, v_0) \dots (u_m, v_m)$, is defined for the m measurement points $p_0, p_1, \dots, p_m$.

Then, because C_{Dist} is the square of the divergence between P and the parameter point,

$$C_{Dist} = \left[\begin{pmatrix} P_0 \\ P_1 \\ \vdots \\ P_m \end{pmatrix} - \begin{pmatrix} B_0(u_0, v_0) & B_1(u_0, v_0) & \dots & B_n(u_0, v_0) \\ B_0(u_1, v_1) & B_1(u_1, v_1) & \dots & B_n(u_1, v_1) \\ \vdots & \vdots & \ddots & \vdots \\ B_0(u_m, v_m) & B_1(u_m, v_m) & \dots & B_n(u_m, v_m) \end{pmatrix} \begin{pmatrix} a_0 \\ a_1 \\ \vdots \\ a_n \end{pmatrix} \right]^2 \quad (3)$$

$$= [P]^T[P] - 2[P][B]^T[a] + [a]^T[B]^T[B][a]$$

$$\therefore C_{Dist} = [f]^T[a] + \frac{1}{2}[a]^T[H][a] \quad (4)$$

$$[f] = -[B]^T[P]$$

$$\text{Linear component: } 1 \times n \text{ (dependent on } X) \quad (5)$$

$$[H] = [B]^T[B]$$

$$\text{Quadratic component: } n \times n \text{ (not dependent on } X) \quad (6)$$

Here, the $[P]^T[P]$ term is excluded because it is a constant, and C_{Dist} is the term used in convergence calculations.

The number of elements for f and H is not dependent on the number of scanned data, m , and is only dependent on the number of control points, n . Because of this, when Eqs. (5) and (6) have been run as initial calculations, optimization calculations can be conducted without the necessity for an enormous amount of scanned data, thus reducing the amount of computation.

In addition, the objective function for the value of G3, C_{Energy} can be defined as follows using the differentiation matrix.

$$C_{Energy} = \frac{\partial^3 S}{\partial u^3} + \frac{\partial^3 S}{\partial v^3} + \frac{\partial^3 S}{\partial u^2 \partial v} + \frac{\partial^3 S}{\partial u \partial v^2} \quad (7)$$

$$= [D_u^3][D_u^3][D_u^3][B][a] + [D_v^3][D_v^3][D_v^3][B][a] + [D_u^2][D_u^2][D_v][B][a] + [D_u][D_u][D_v^2][B][a]$$

The final objective function, C , is

$$C = C_{Dist} + \lambda C_{Energy} \quad \lambda: \text{Constant} \quad (8)$$

Next, the constraint conditions are determined. Now, the value of G3 in the uv direction is assumed to be greater than b_L and less than b_H . The constraint conditions for each point at this time can be expressed as follows, using the differentiation matrix:

$$b_L^{u3} < \frac{\partial^3 S}{\partial u^3} < b_H^{u3}, \quad b_L^{v3} < \frac{\partial^3 S}{\partial v^3} < b_H^{v3}, \quad b_L^{u2v1} < \frac{\partial^3 S}{\partial u^2 \partial v} < b_H^{u2v1}, \quad b_L^{u1v2} < \frac{\partial^3 S}{\partial u \partial v^2} < b_H^{u1v2} \quad (9)$$

Here, b^{u3} , b^{v3} , b^{u2v1} , and b^{u1v2} at both ends are the partially differentiated G3 values for each rank in the u, v space, and vary with the degree of stiffness specified. In concrete terms, the upper limit value of the G3 constraint is absolutely and quantitatively defined, and the lower

limit value of the G3 constraint is determined on the basis of how stiff the surface will be made in comparison to the results obtained using the upper limit value.

Linear programming is often used in convergence calculations for conditional linear matrices. However, in this case, because quadratic terms appeared in the objective function, extended quadratic programming was used for the quadratic expression. OOQP⁽⁴⁾ was used for the algorithms to solve the quadratic programming tasks.

As Eqs. (4) and (9) show, the objective function, C , and the constraint condition, A , can be expressed as a general expression as follows. Below, in accordance with the sign system used in OOQP⁽⁴⁾, the matrix expression $[X]$ has been simplified to X .

$$C = \frac{1}{2} a^T H a + f^T a \quad (10)$$

$$A a > b \quad (11)$$

a : Solution

H : Objective function (quadratic component)

f : Objective function (linear component)

A : Constraint condition (computational expression)

b : Constraint condition (value)

In addition, taking the G2 connection constraint conditions expressed in Eq. (19) below and other factors into consideration, constraint conditions with equal signs are defined as follows:

$$D a = d \quad (12)$$

D : Constraint condition (computational expression)

d : Constraint condition (value)

Satisfying the conditions of Eqs. (11) and (12), the minimal solution of Eq. (10), a , is found. When there are no constraint conditions,

$$dC/d\mathbf{a} = 0 \quad \therefore H\mathbf{a} = -f \quad (13)$$

for the optimal solution, $\mathbf{a}$. When constraint conditions exist, this is not always satisfied. Therefore, taking into consideration new unknowns for the respective constraint conditions, y , z , and s , L is defined using the equation below. Here, because s and z form an inequality, they are given positive values.

$$L = \frac{1}{2} a^T H a + f^T a - y(Da - d) - z(Aa - s - b) \quad (14)$$

$$s \geq 0 \perp z \geq 0$$

Here, the first and second terms express the objective functions, the third term expresses the equality constraint condition, and the fourth term expresses the inequality

constraint condition. At this time, the following conditional expressions are formed using the optimal solution, $\mathbf{a}$.

$$dL/d\mathbf{a} = 0 \quad \therefore H\mathbf{a} - D^T y - A^T z = -f \quad (15)$$

$$dL/dy = 0 \quad \therefore A^T \mathbf{a} = b \quad (16)$$

$$dL/dz = 0 \quad \therefore C^T z - s = d \quad (17)$$

$$dL/ds = 0 \quad \therefore sz = 0 \quad (18)$$

In this method, these conditional expressions are solved using the Newton method in order to derive $\mathbf{a}$.

3.2. Merge Technology

Completed Class A data is made up of multiple surfaces which display continuity of position, G0, tangency, G1, and curvature, G2. The target for the new system was to produce this continuity automatically.

Using the NURBS equation, the unique definition of numbers in the terminal section helps to enable continuity to be obtained as desired. However, this method produces prominent disturbance in the interior of the surface, and is unsuitable for Class A surfaces. In the new system, therefore, constraint conditions for continuity between surfaces were set in addition to constraint conditions for intelligent fitting surfaces.

The objective function defined in Eq. (8) is not limited to single surfaces, but can also be used in calculations for multiple surfaces.

The simultaneous calculation of the control points for surface 1, $a_0, a_1 \dots a_n$, and the control points for surface 2, $b_0, b_1 \dots b_n$, will now be considered. To help ensure G2 continuity in the u direction for the points defined by (u_0, v_0) on surface 1 and the control points (u_1, v_1) on surface 2, the following should hold:

$$\frac{\partial^2 S_1}{\partial u_1^2} - \frac{\partial^2 S_2}{\partial u_2^2} = 0 \quad (19)$$

Here, by adding the conditional expression Eq. (19) to the intelligent fitting surfaces obtained from Eq. (8) with $a_0, a_1 \dots a_n, b_0, b_1 \dots b_n$ as the unknowns, it is possible to obtain two surfaces with continuity of curvature at their boundaries.

4. Technologies for Increased Speed and Accuracy of Fillet Surfacing

The shapes used in recent automotive styling have increased in complexity, and the number of surfaces

employed has approximately doubled against five years previously, making an increase in the number of man-hours involved in the creation of Class A data an issue. Because the accuracy of surface data is important in Class A data in addition to the reproduction of styling shapes, it is necessary to balance increased speed and increased accuracy in order to reduce the number of man-hours involved in its creation.

As Fig. 6 shows, a fillet is a shape that rounds the angle formed by two sets of surfaces. In automotive styling, fillet surfaces represent approximately 10% of the total number of Class A surfaces. The new system set out to increase the speed of computational processing of fillet surfaces and the accuracy of fillet surface quality.

At Honda, the fillets that obtain G2 continuity with support surfaces are mainly used. To create such fillets, first a group of curves called ribs that obtain G2 continuity with the support surface is calculated, as shown in Fig. 7. And then the fillet surface is generated using these ribs. In the new system, increased speed was achieved through the modification of the ribs generation method, and accuracy was increased through the modification of the surfacing method.

4.1. Technology for Increased Speed

As Fig. 8 shows, the ribs that obtain G2 continuity with the support surfaces are generated as fifth-degree 1-segment B-spline curves with first-order and second-order

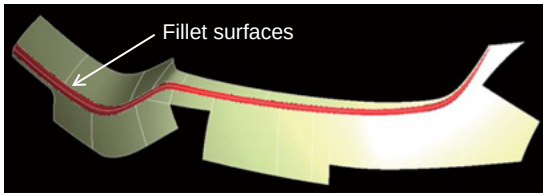


Fig. 6 Sample of fillet

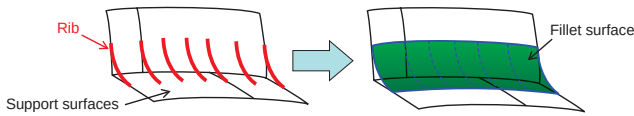


Fig. 7 Sample of rib

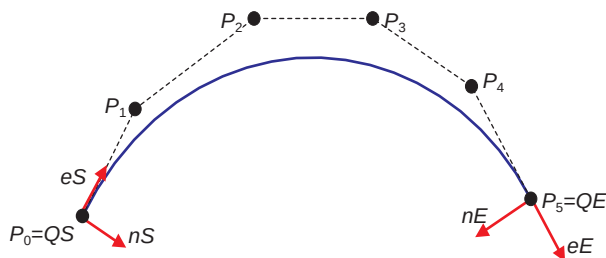


Fig. 8 Control points of 5 degree rib curve

differential vectors as input conditions at both ends. The first-order differential vectors express tangency, and the second-order differential vectors express curvature.

In the previous method, differentials were obtained from the intersection curves between the normal sectional planes of spine curve and the support surfaces in order to find the tangency and curvature of the end points. Because the calculations for the intersections were convergence calculations, this presented the issue of a high calculation time for high-quality fillet surfaces. For the new system, a method was developed for high-speed calculation of the curvature of the rib end points without calculation of intersection, using the surface unit normal vectors and the sectional plane unit normal vectors of the rib end points on the support surface and the normal curvature of the support surface.

For the curve $C(t)$, the unit tangent vector is termed $e(t)$, the unit principal normal vector is termed $n(t)$, and the curvature function is termed $\kappa(t)$.

With $0 \leq t \leq 1$ as a parameter condition and P_0, P_1, P_2, P_3, P_4 , and P_5 as the control points, a fifth-degree Bezier curve, $C(t)$, is expressed as follows:

$$C(t) = (1-t)^5 P_0 + 5(1-t)^4 t P_1 + 10(1-t)^3 t^2 P_2 + 10(1-t)^2 t^3 P_3 + 5(1-t) t^4 P_4 + t^5 P_5 \quad (20)$$

The first- and second-order differential vectors for the curve are expressed as:

$$\Delta P_i = P_{i+1} - P_i \quad (21)$$

$$\Delta^2 P_i = \Delta(\Delta P_i) = \Delta P_{i+1} - \Delta P_i = P_{i+2} - 2P_{i+1} + P_i \quad (22)$$

Using Eqs. (21) and (22), the following expressions can be obtained:

$$\frac{dC}{dt} = 5 \{ (1-t)^4 \Delta P_0 + 4(1-t)^3 t \Delta P_1 + 6(1-t)^2 t^2 \Delta P_2 + 4(1-t) t^3 \Delta P_3 + t^4 \Delta P_4 \} \quad (23)$$

$$\frac{d^2 C}{dt^2} = 20 \{ (1-t)^3 \Delta^2 P_0 + 3(1-t)^2 t \Delta^2 P_1 + 3(1-t) t^2 \Delta^2 P_2 + t^3 \Delta^2 P_3 \} \quad (24)$$

The first- and second-order differential vectors of the start and end points are:

$$\left. \frac{dC}{dt} \right|_{t=0} = 5(\Delta P_0), \quad \left. \frac{dC}{dt} \right|_{t=1} = 5(\Delta P_4) \quad (25)$$

$$\left. \frac{d^2 C}{dt^2} \right|_{t=0} = 20(\Delta P_1 - \Delta P_0), \quad \left. \frac{d^2 C}{dt^2} \right|_{t=1} = 20(\Delta P_4 - \Delta P_3) \quad (26)$$

The first-order differential vector is calculated from the tangent planes at the end points of the ribs on the surface and the normal sectional plane of the spine curve, and P_0 and P_4 in Eq. (25) are determinant.

A Frenet equation⁽⁵⁾ calculated using the arc length parameter expresses the fact that the second-order

differential vector and the first-order differential vector are orthogonal to each other⁽⁶⁾. The magnitude of the second-order vector is determined using the method discussed below. P_2 and P_3 in Eq. (26) are determinant.

Figure 9 shows the method of calculation of the curvature at the end point of a rib.

Using the following definitions,

n : Unit normal vector of support surface at end point of rib

m : Unit normal vector of sectional plane

e : Rib unit tangent vector at end point of rib (calculated from intersection of tangent plane and sectional plane)

α : Angle formed by n and m

κ_n : Normal curvature in e direction at end point of rib

κ : Curvature of cross-section at end point of rib

κ can be calculated using the following equation without explicit determination of the intersection curves:

$$\kappa = \left| \frac{\kappa_n}{\sin \alpha} \right| \quad (27)$$

However, when $\alpha \equiv 0$, the rib cannot be determined because the sectional plane and the tangent plane are identical.

The normal curvature at points on the surface (Fig. 10) is calculated using the following method.

The surface equation is defined as $S(u, v)$ and the unit normal vector is defined as $n(u, v)$. The introduction of e , the unit tangent vector at point p on the surface, determines the sectional plane that includes

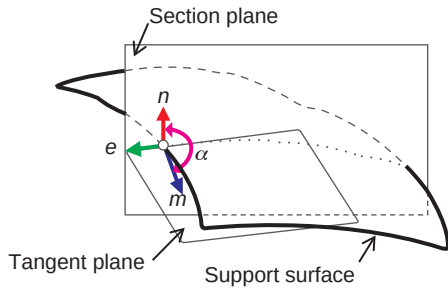


Fig. 9 Method of calculation of curvature at end point of rib

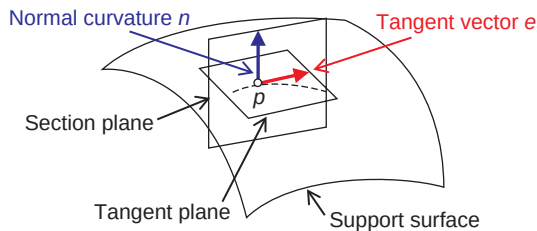


Fig. 10 Normal curvature of surface

e and n . The surface section in this plane is found, and the curvature of the section at point p (considered to be allocated a positive sign when the center of the curvature is on the n side, and a negative sign when the center of the curvature is on the side opposite the n side) is termed the normal curvature at point p on the surface in the e direction. If the tangent vector, e , is made to describe a circuit on the tangent plane around, the normal curvature generally also changes. The maximum and minimum values of the normal curvature are termed the principal curvature. Because the tangent vectors that impart principal curvature are orthogonal to each other, they are termed the principal direction. These calculations are conducted as follows. $\langle \cdot, \cdot \rangle$ represents the inner product of the vectors, and the partial differential at the parameters u, v is expressed as $S_u, S_v, S_{uu}, S_{uv}, S_{vv}$. First, coefficients of the first fundamental form

$$E = \langle S_u, S_u \rangle, F = \langle S_u, S_v \rangle, G = \langle S_v, S_v \rangle \quad (28)$$

and coefficients of the second fundamental form

$$L = \langle S_{uu}, n \rangle, M = \langle S_{uv}, n \rangle, N = \langle S_{vv}, n \rangle \quad (29)$$

are calculated for the position of the point on the surface at which normal curvature will be calculated.

The unit tangent vector, e , is expressed as

$$e = \xi S_u + \eta S_v \quad (30)$$

However,

$$\langle e, e \rangle = \langle \xi S_u + \eta S_v, \xi S_u + \eta S_v \rangle = E\xi^2 + 2F\xi\eta + G\eta^2 = 1 \quad (31)$$

If the principal curvature is termed κ_j , and the corresponding principal direction is termed $e_j = \xi_j S_u + \eta_j S_v$ ($j = 1, 2$), then the principal curvature and principal direction are resolved into the following equation for the determination of eigenvalues and eigenvectors.

$$\frac{1}{EG-F^2} \begin{bmatrix} G & -F \\ -F & E \end{bmatrix} \begin{bmatrix} L & M \\ M & N \end{bmatrix} \begin{bmatrix} \xi_j \\ \eta_j \end{bmatrix} = \kappa_j \begin{bmatrix} \xi_j \\ \eta_j \end{bmatrix} \quad (32)$$

Solving this eigenvalue equation is taken as giving the principal directions and the principal curvature. Because the two principal directions are the orthonormal base of the tangent plane, they can be expressed as

$$e = (\cos \theta) e_1 + (\sin \theta) e_2 \quad (33)$$

At this time, the following Euler formula is generated between the normal curvature in the e direction, κ_n , and the principal curvature.

$$\kappa_n = (\cos \theta)^2 \kappa_1 + (\sin \theta)^2 \kappa_2 \quad (34)$$

Using this formula, the normal curvature in the e direction, κ_n , can be calculated. Finding the sectional curvature at the end points of the ribs from Eq. (27) and finding the fifth-degree 1-segment B-spline curve makes it possible to determine the ribs at high speeds without intersection calculations.

4.2. Technology for Achievement of High Level of Accuracy

Using the previous technology, continuity between the intersection curves of normal sectional planes and support surfaces and intersection curves of normal sectional planes and fillets was sought in order to achieve continuity between support surfaces and fillet surfaces. Because the intersection calculations were convergence calculations, despite the fact that considerable time was expended on the calculations, it was not possible to ensure continuity other than at the position of intersections. In the new system, G2 continuity between support surfaces and fillet surfaces has been realized through the employment of phased Gordon surfacing using cross-boundary derivatives (CBD) (Fig. 11), which provide information concerning the differential values of peripheral surfaces.

Fillet surfaces are generated in the following phases.

- (1) In order to achieve G1 continuity with support surfaces, the local surface produced by the end rib $U_{0,0}(u)$ and two proximate ribs, $U_{0,1}(u)$ and $U_{0,2}(u)$, is focused on; the first-order CBD curve is found using the partial differential in the v direction, and twist vector matching termed global deformation⁽⁷⁾ is applied based on the tangent vector of the support surface.
- (2) In order to achieve G2 continuity with support surfaces, the local surface produced by the end rib, $U_{0,0}(u)$, one reference rib on the inner part of the surface, $U_{0,1}(u)$, and the previously determined first-order CBD curve is focused on, and the second-order CBD curve is determined from the second-order partial differential.
- (3) A G0 Gordon surface is generated based on the first-order CBD curve and second-order CBD curve, without consideration of continuity⁽⁷⁾.

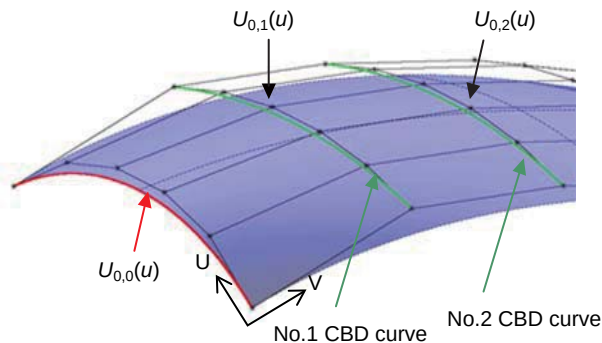


Fig. 11 Cross-boundary derivative curves No.1 and No.2 at $U_0(u)$

- (4) A first-order CBD curve is again obtained from the G0 Gordon surface, and global deformation is again applied in order to achieve G2 continuity with the support surface.
- (5) A Gordon surface is generated from the first-order CBD curve, the second-order CBD curve, the first-order CBD curve for the G0 Gordon surface, the R tangent curve, and the ribs, then the Gordon surface becomes the output.

4.2.1. Determination of first-order CBD following end rib

In the case of the starting end rib, the second-order Lagrange interpolating polynomial $\xi_{0,s}(v)$, $\xi_{1,s}(v)$, $\xi_{2,s}(v)$, which passes a total of three ribs, the starting end rib, $U_{0,0}(u)$, and the two proximate end ribs for reference, $U_{0,1}(u)$ and $U_{0,2}(u)$, is focused on.

From the local surface derived from

$$T(u,v) = U_{0,0}(u)\xi_{0,s}(v) + U_{0,1}(u)\xi_{1,s}(v) + U_{0,2}(u)\xi_{2,s}(v) \quad (35)$$

the starting end CBD is calculated:

$$\frac{\partial T}{\partial v}(u, b_0) = U_{0,0}(u)\xi'_{0,s}(b_0) + U_{0,1}(u)\xi'_{1,s}(b_0) + U_{0,2}(u)\xi'_{2,s}(b_0) \quad (36)$$

At this time, the corner twist vector is determined from the support surface, and twist matching of the tangent vector of the end point of the first-order CBD control curve is necessary. This is termed global deformation of the CBD control curve. Because G2 twist matching with the support surface at the end point of the end rib is considered in the new system, global deformation is applied for twist matching of the first-order CBD control curves and to transform it into a twist vector⁽⁷⁾.

A first-order CBD control curve is calculated in the same way for the finishing end rib, but in this case it is differentiated from the direction of the finishing end.

4.2.2. Determination of second-order CBD following end rib

A method of interpolation with 1 X 2 degree Coons surface will be formulated using the end rib, an inside rib of the end rib, and the first-order CBD function along the end rib. A second-order CBD function is calculated from this Coons surface, $T(u, v)$. The Coons surface, $T(u, v)$, can be expressed as follows:

$$T(u,v) = T_1(u,v) + T_2(u,v) - T_3(u,v) \quad (37)$$

Defining these terms as follows

$$T_1(u,v) = [U_0(u), D_s(u), U_1(u)] \begin{bmatrix} \lambda_0(v) \\ \lambda_1(v) \\ \lambda_2(v) \end{bmatrix} \quad (38)$$

$$T_2(u,v) = [\mu_0(u), \mu_1(u)] \begin{bmatrix} V_0(v) \\ V_1(v) \end{bmatrix} \quad (39)$$

$$T_3(u,v)=[\mu_0(u), \mu_1(u)] \begin{bmatrix} U_0(a_0), D_s(a_0), U_1(a_0) \\ U_0(a_m), D_s(a_m), U_1(a_m) \end{bmatrix} \begin{bmatrix} \lambda_0(v) \\ \lambda_1(v) \\ \lambda_2(v) \end{bmatrix} \quad (40)$$

with $\mu_j(u)$ as the interpolation function in the u direction, $\lambda_j(v)$ as the interpolation function in the v direction, $D_s(u)$ as the first-order CBD control curve on the starting end side, u as the interval $[a_0, a_m]$, and v as the interval $[b_0, b_n]$, and determining $\frac{\partial^2 T}{\partial v^2}(u, b_0)$, produces a second-order CBD function.

A second-order CBD control curve is calculated in the same way for the finishing end rib, but in this case it is differentiated from the direction of the finishing end.

4.2.3. Generation of G0 Gordon surfaces

Gordon surfaces are generated from the R tangent curve, the ribs, and the first-order and second-order CBD functions following the end rib without consideration of continuity with the support surfaces⁽⁷⁾. At this time, the interpolation function in the v direction is considered as a fifth-degree C4 spline function.

$$\tilde{T}(u,v) = \tilde{T}_1(u,v) + \tilde{T}_2(u,v) - \tilde{T}_3(u,v) \quad (41)$$

However, with $D_e(u)$ as the first-order CBD control curve on the finishing end side, $E_s(u)$ as the second-order CBD control curve on the starting end side, and $E_e(u)$ as the second-order CBD control curve for the finishing end,

$$\tilde{T}_1(u,v) = [U_0(u), D_s(u), E_s(u), U_1(u), U_2(u), \dots, U_{n-2}(u), U_{n-1}(u), E_s(u), D_s(u), U_n(u)] \begin{bmatrix} \lambda_0(v) \\ \lambda_1(v) \\ \vdots \\ \lambda_{n+1}(v) \end{bmatrix} \quad (42)$$

$$\tilde{T}_2(u,b_0) = [\mu_0(u), \mu_1(u)] \begin{bmatrix} V_0(v) \\ V_1(v) \end{bmatrix} \quad (43)$$

$$\tilde{T}_3(u,v) = [\mu_0(u), \mu_1(u)] \begin{bmatrix} U_0(a_0), D(a_0), E(a_0), U_1(a_0), U_2(a_0), \dots, U_{n-2}(a_0), U_{n-1}(a_0), E(a_0), D(a_0), U_n(a_0) \\ U_0(a_n), D(a_n), E(a_n), U_1(a_n), U_2(a_n), \dots, U_{n-2}(a_n), U_{n-1}(a_n), E(a_n), D(a_n), U_n(a_n) \end{bmatrix} \begin{bmatrix} \lambda_0(v) \\ \lambda_1(v) \\ \vdots \\ \lambda_{n+1}(v) \end{bmatrix} \quad (44)$$

4.2.4. Determination of first-order CBD following each internal rib from the G0 Gordon surface

First-order CBD control curves for the internal ribs are determined from the G0 Gordon surface, and global deformation is applied in order to achieve G2 continuity with the support surfaces.

4.2.5. Generation of G1 Gordon surfaces

Using fifth-degree C3 interpolation, Gordon surfaces are generated from the first-order CBD functions following the internal ribs which have undergone correction as described above, the R tangent curve, the ribs, and the first- and second-order CBD functions following the end rib. Expressed as $\tilde{T}(u, v)$, these surfaces

are termed G1 Gordon surfaces, and are treated as output surfaces.

$$\tilde{T}(u,v) = \tilde{T}_1(u,v) + \tilde{T}_2(u,v) - \tilde{T}_3(u,v) \quad (45)$$

However, with $D_i(u)$ as the first-order CBD control curves following the internal ribs,

$$\tilde{T}_1(u,v) = [U_0(u), D_0(u), E_0(u), U_1(u), D_1(u), U_2(u), D_2(u), \dots, U_{n-2}(u), D_{n-2}(u), U_{n-1}(u), E_1(u), D_1(u), U_n(u)] \begin{bmatrix} \lambda_0(v) \\ \lambda_1(v) \\ \vdots \\ \lambda_{n+1}(v) \end{bmatrix} \quad (46)$$

$$\tilde{T}_2(u,b_0) = [\mu_0(u), \mu_1(u)] \begin{bmatrix} V_{10}(v) \\ V_{11}(v) \end{bmatrix} \quad (47)$$

$$\tilde{T}_3(u,v) = [\mu_0(u), \mu_1(u)] \begin{bmatrix} U_0(a_0), D(a_0), E(a_0), U_1(a_0), U_2(a_0), D(a_0), \dots, U_{n-2}(a_0), D_{n-2}(a_0), U_{n-1}(a_0), E(a_0), D(a_0), U_n(a_0) \\ U_0(a_n), D(a_n), E(a_n), U_1(a_n), U_2(a_n), D(a_n), \dots, U_{n-2}(a_n), D_{n-2}(a_n), U_{n-1}(a_n), E(a_n), D(a_n), U_n(a_n) \end{bmatrix} \begin{bmatrix} \lambda_0(v) \\ \lambda_1(v) \\ \vdots \\ \lambda_{n+1}(v) \end{bmatrix} \quad (48)$$

5. Global Deformation Technology

The previous method of recreating Class A surface data from the beginning, when clay models were updated frequently in the quest for high-quality styling designs, meant that changes to the data could not keep pace with design changes. In order to increase the efficiency of the generation of Class A data, the existing approach to deforming and correcting Class A surfaces was focused on.

In the previous method of correcting surfaces, obtaining a target shape involved, first, a repeated process of adjusting the mathematical parameters of all the individual surfaces and expressing the desired shapes, following which it was necessary to make fine adjustments to the surfaces in order to connect them with continuity of curvature. These were time-consuming procedures. In the new system, the use of a method of correction of surface shapes that makes it possible to directly input shape targets based on the aesthetic judgment of styling designers has helped to enable the achievement of a more intuitive and efficient data correction process. This is termed “target-driven modeling.” Technologies were developed that make it possible to input a larger number of surfaces at the same time while preserving existing continuity, thus outputting a group of surfaces that displayed continuity. Highlight control and curvature control will be discussed here as examples of these technologies.

5.1. Highlight Control

Highlight control applies the concept of target-driven modeling to two-dimensional highlights, one of the standards employed in the evaluation of the quality of

Class A data. This function makes it possible, by specifying a target highlight for an existing group of surfaces, to deform the surfaces into a shape that will produce the target highlight. The upper panel in Fig. 13 shows the input surface group and the target highlight intended by the styling designer, and the lower panel shows the output result. The blue lines in the images are the highlights on the surface rendered visible. In this example, the surfaces were deformed in order to make the highlight shown by the yellow line move into the position specified by the white line.

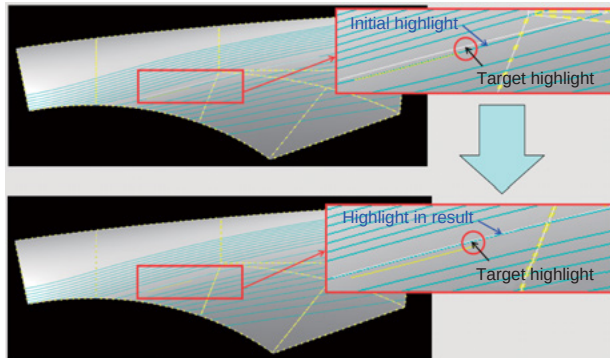


Fig. 13 Input and result of highlight control

5.1.1. Two-dimensional highlights

Figure 14 shows a representation of the definition of a two-dimensional highlight. A two-dimensional highlight is a highlight generated on a surface for evaluation of the styling shape of the surface. A group of planes in a fixed direction are defined for the surface to be evaluated, and a line intersecting these planes is generated. The points at which the tangent vectors of the intersecting line display the same orientation are isolated and connected. This is defined as a curve on the surface to be obtained.

5.1.2. Highlight control method

As indicated in the preceding section, highlights are generated by the connection of points on a surface that have the same tangent direction. Given this, controlling a highlight on a surface involves moving the highlight from the existing highlight line to the target highlight line while preserving the tangent vector. However, as Fig. 15 shows, when the tangent vector of the highlight line is preserved with the existing continuity at the edges of the surface fixed, the surface shapes around the highlight represent the opposite tendency due to over-constraints on the surface operations.

In order to resolve this issue, the following degrees of freedom of surface deformation were introduced in addition to the basic concept of preserving the tangent

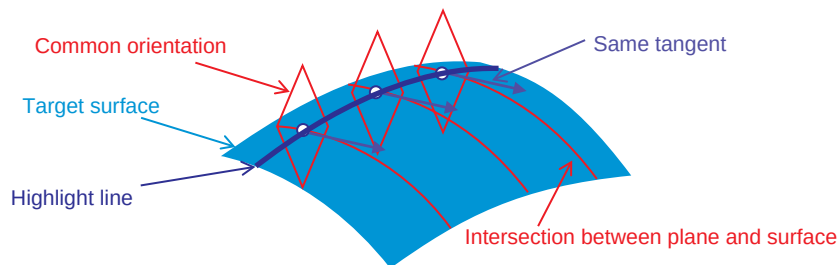


Fig. 14 Definition of 2D highlight

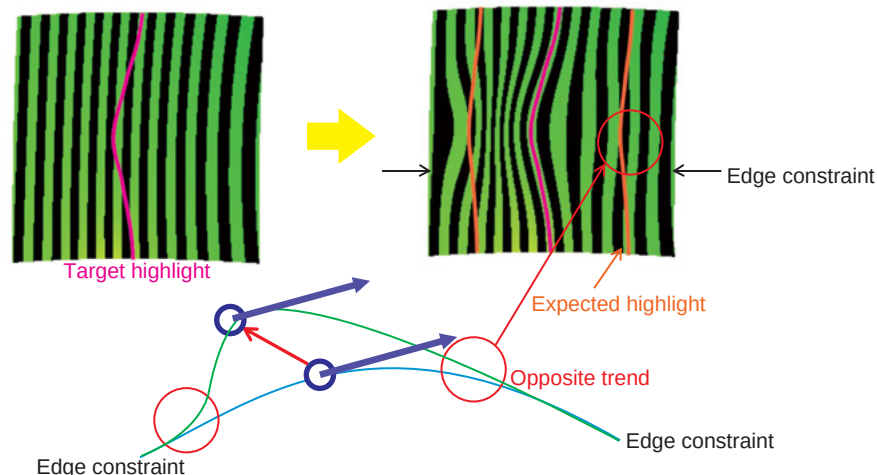


Fig. 15 Issue from over-constraints

vector of the highlight line:

- (1) Provision of degree of freedom to normal direction of surface while preserving tangent vector
- (2) Provision of degree of freedom of position in relation to direction of view in highlight operations

This is shown in Fig. 16. Because the input target highlight line in this function specifies the transformation of the trend of the surface shape, the degrees of freedom described above make it possible to realize the target highlight without any negative effects. The use of this method has resolved the technological issues associated with highlight control.

5.2. Curvature Control

Curvature control is a function that makes it possible to deform an existing curve shape by inputting a target line for a curvature plot to modify an existing curve. In the previous system, the control points and pass points of the curve were adjusted one by one in a repeated process in order to obtain the target curve. As Fig. 17 shows, the purpose of the curvature control function is, based on the concept of target-driven modeling, to help enable the direct deformation of the curve shape to obtain the target curvature plot by inputting the existing curve, C , and the target curvature plot, TF .

5.2.1. Curvature control method

Assuming, for an arbitrary point P on curve C in Fig. 17, that the curvature rate $\underline{\kappa}$ that satisfies the target curvature plot TF occurs at the point of intersection between the normal vector of point P and TF , termed Pf ,

$$\underline{\kappa} = (P - Pf) / \text{Drawing scale of curvature plot} \quad (49)$$

Expressing this for the entire curve using parameter s for curve C ,

$$\underline{\kappa}(s) = (C(s) - TF) / \text{Drawing scale of curvature plot} \quad (50)$$

Terming the starting point of the initial curve C P_0 and the tangent vector of the starting point T_0 , the curve CT , which satisfies the target curvature plot, can be found using

$$\theta(s) = \text{Angle}[T_0] + \int_{[0,s]} \underline{\kappa}(u) \frac{du}{ds} ds \quad (51)$$

$$CT(s) = P_0 + \int_{[0,s]} \{\cos[\theta(u)], \sin[\theta(u)]\} \frac{du}{ds} ds \quad (52)$$

5.2.2. Constraint conditions at start and end points

However, using Eq. (52), it is not possible to specify the position of the end point of the resulting curve CT . In Class A modeling, it is important to preserve not only the position of start and end points of an existing curve, but also their tangent vectors and curvatures.

For example, terming the curvature of the resulting curve CT κ_c , in order to preserve the tangent vectors of the start and end points of the curve it is necessary to satisfy the condition

$$\int_0^u \kappa_c(u) du = \int_0^u \underline{\kappa}(u) du \quad (53)$$

In addition, in order to preserve the curvature of the start and end points of the curve it is necessary to satisfy the conditions

$$\kappa_c(u_a) = \underline{\kappa}(u_a) \quad (54)$$

$$\kappa_c(u_b) = \underline{\kappa}(u_b) \quad (55)$$

Figure 18 conceptualizes Eqs. (53) and (55). In order to preserve the tangent vectors of the start and end points

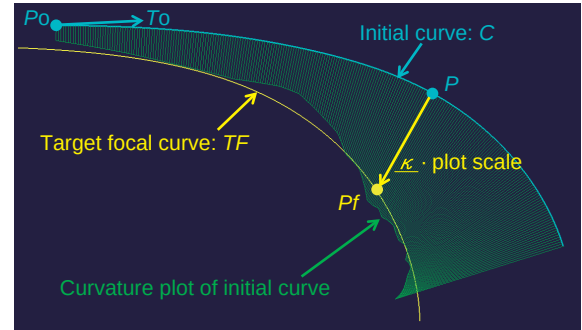


Fig. 17 Input for curvature control

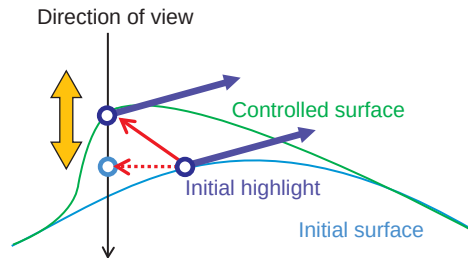
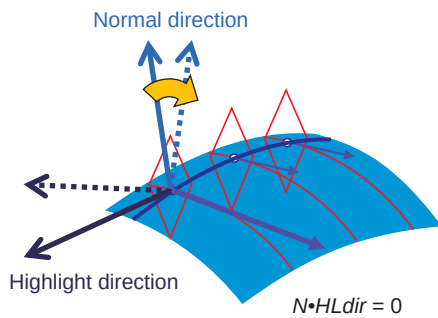


Fig. 16 With given freedom

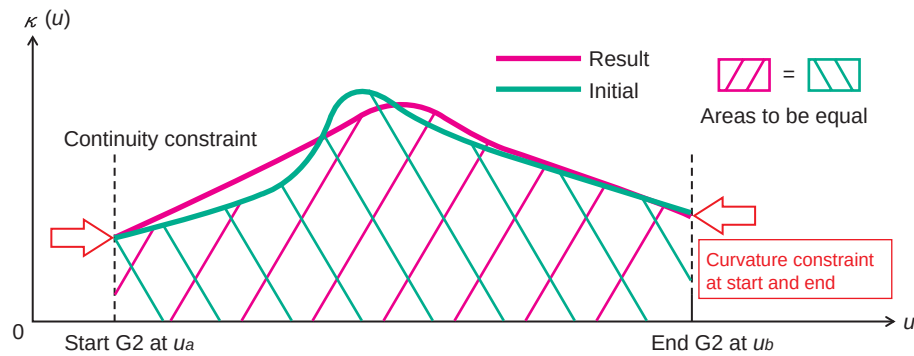


Fig. 18 Constraint of integrated area

of the existing curve, the integrated areas of the initial curve and the resulting curve are to be equal, with the position parameters for the curves specified as $u [u_a, u_b]$ and the curvature at those positions as $\kappa(u)$.

However, because the target curvature plot, TF , is drawn and input by the user, it is challenging to completely satisfy the constraint conditions discussed above. Correction of the curvature $\kappa(u)$, obtained from Eq. (49), is therefore necessary in order to satisfy the constraint conditions expressed by Eqs. (53) to (55). Using a dynamically defined scalar function, $\varphi(u)$, the new system conducts this correction as follows:

$$\kappa_m(u) = \kappa(u) \times \varphi(u) \quad (56)$$

$\kappa_m(u)$ is substituted for the curvature obtained from Eq. (49). By this means, the constraint conditions can be satisfied at the start and end points of the curve, and curvature control is realized with the continuity of shape necessary in Class A modeling.

6. Effects of New Technologies

6.1. Polygon Processing Technologies

The following results have been obtained in the polygon processing technologies in the new system.

- (1) Efficient polyline section calculation using half-edge data structure has been realized. Figure 19 shows calculation results.
- (2) The division of projected curves into segments has helped to enable more accurate projection calculations. Figure 20 shows calculation results.
- (3) The use of an original method for the efficient generation of initial patches has helped to enable the realization of high-speed and robust automatic hole-filling. Figure 3 shows the calculation process and calculation results.
- (4) The use of an original intersection calculation method for high-speed space division of mesh face sets has helped to enable the realization of more efficient calculation of distance between polygon mesh and geometric shapes. Figure 21 shows a visualization of calculation results.

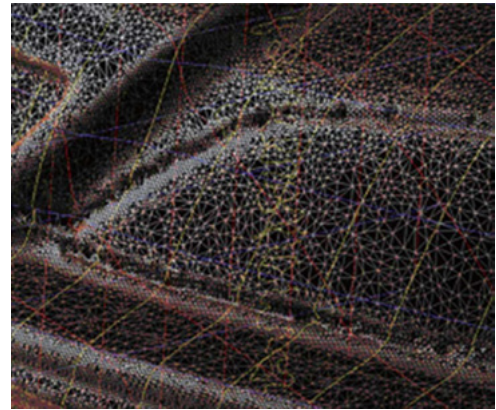


Fig. 19 Creation of polyline-section on scanned data

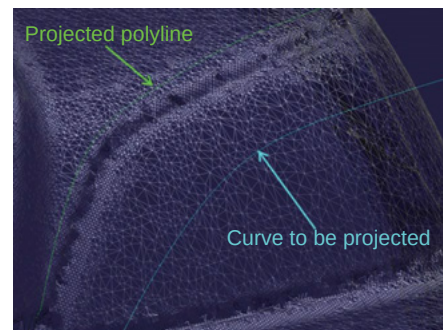


Fig. 20 Projection curve on scanned data

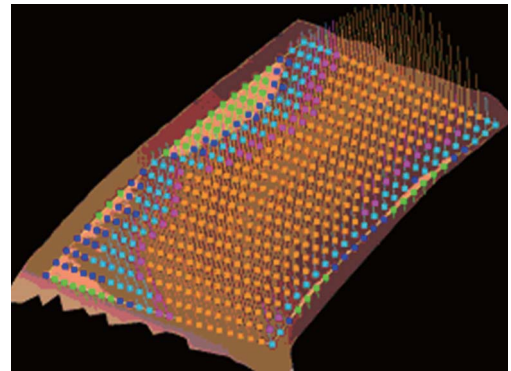


Fig. 21 Mesh-surface distance map

Table 1 Fillet performance

	Previous system	New system
Sample of fillet 1	27 min.	15 sec.
Sample of fillet 2 (Fig. 22)	52 sec.	5 sec.
Sample of fillet 3	34 sec.	3 sec.

6.2. Fitting Technologies

The following results have been obtained through the use of fitting technologies in the new system⁽¹⁾.

- (1) The use of intelligent fitting technologies has helped to ensure shape quality for individual fitting surfaces.
- (2) The application of a merge technology using fitting technologies has helped to enable surface connection with continuity between multiple surfaces.

6.3. Fillet Technologies

The use of fillet technologies in the new system has simultaneously increased the speed of computation and the accuracy of the output shapes⁽¹⁾.

- (1) Extensive revision of the rib calculation algorithm has increased the speed of computation. Table 1 compares times for the generation of fillets in the previous and the developed systems. For all samples, times have been reduced by 90% or more in the new system. Figure 22 shows the most complex shape sample.
- (2) The use of Gordon surfacing in the final stage of generation of fillet surfaces has enabled the creation of fillets with G2 continuity with support surfaces. In addition, the use of phased Gordon surfacing has increased the shape quality of the fillet surfaces themselves. Figure 23 shows a visualization of the Gauss curvature of fillet shapes produced by the previous and the developed system on

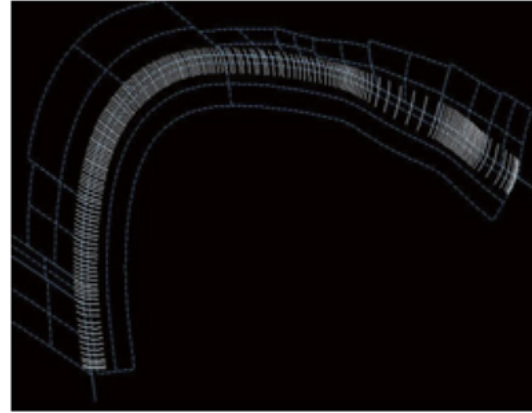


Fig. 22 Sample of fillet 2

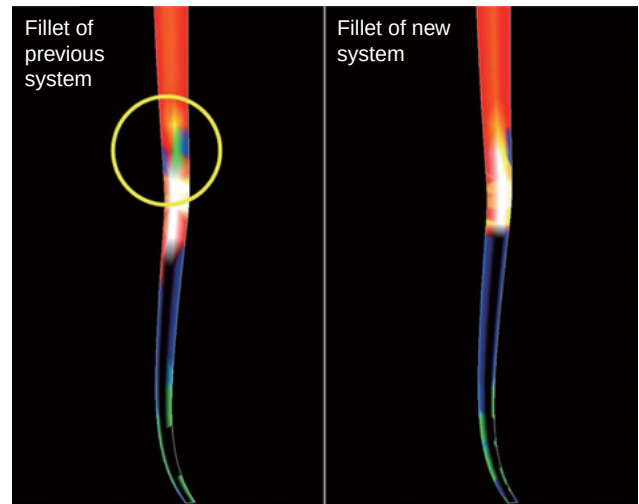


Fig. 23 Visualization of gauss curvature

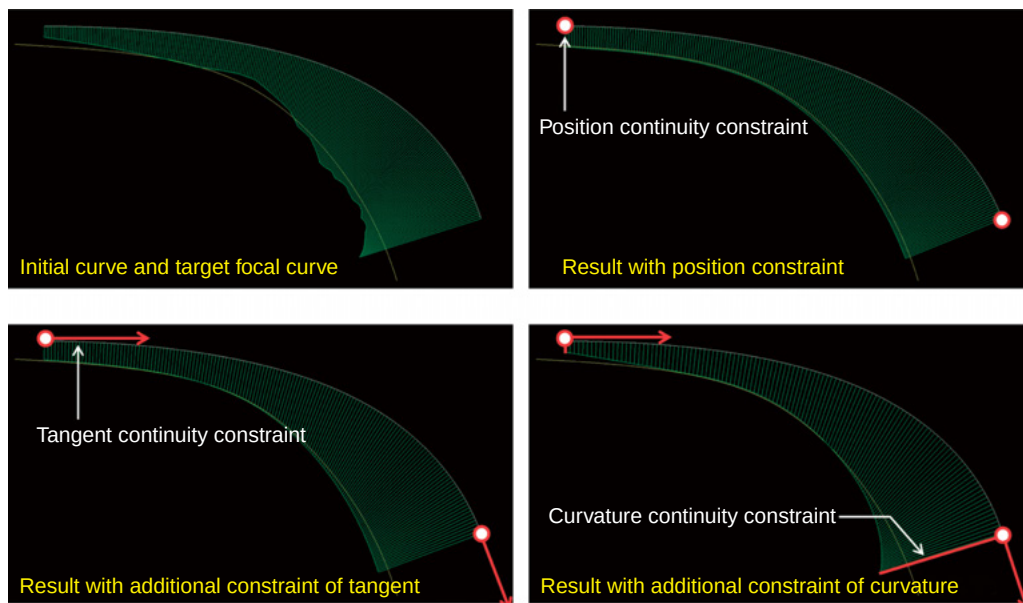


Fig. 24 Input and results of curvature control with various continuity constraints

the basis of identical input. While the fillet surface output by the previous system displays sudden changes in Gauss curvature, a high level of shape quality is achieved in the fillet surface generated by the new system.

6.4. Global Deformation Technologies

Highlight control and curvature control are examples of the realization of target-driven modeling through the application of global deformation technologies⁽¹⁾.

- (1) The highlight control function makes it possible to quickly modify a styling shape composed of multiple surfaces to a target shape simply by specifying a target highlight. The example shown in Fig. 13 would have taken 60 minutes using the previous system, but was conducted by the new system in just 43 seconds.
- (2) The curvature control function makes it possible to deform an existing curve simply by specifying a target focal curve for a curvature plot and continuity conditions for the start and end points of the curve. Figure 24 shows the results output for the same input curve and target curvature plot when only the continuity conditions of the start and end points of the curve are varied. The use of this technology has reduced the time employed in this procedure to 17% in the best cases against the previous method.

7. Conclusion

In order to realize the target of reducing the man-hours necessary for the creation of Class A surface data by 50%, the project discussed in this paper focused on three phases of the modeling process up to the stage of completion of the Class A surfaces, and sought to develop CAD functions which would considerably increase work efficiency in each phase.

- (1) In the phase of generation of fitting surfaces from scanned data from clay models, the use of an original polygon work bench has helped to enable high-speed and robust processing of high-volume polygon data, creating a foundation for the process of fitting surfaces to scanned data. In addition, the surface fitting function realizes surface generation that maintains accuracy in relation to the scanned data while satisfying the quality conditions for basic Class A surfaces, and has increased efficiency at the initial stage of Class A data generation.
- (2) With regard to achieving increased speed and accuracy in Class A surfacing functions, efficiency and data quality have been increased for central functions in the process of generation of Class A data. For example, the application of an original algorithm in the frequently used fillet generation function has increased the speed and accuracy of computation.
- (3) In the phase of correction of Class A data, the introduction of the new concept of target-driven modeling has increased the efficiency of the complex process of responding to adjustments of styling orientation and changes in designs.

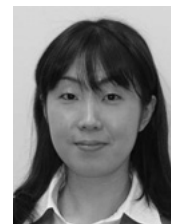
References

- (1) Shinohara, T., Sunohara, K., Sakai, S., Katsube, H., Ide, D., Hara, M., Kaneda, T.: Development of Class A Surfacing CAD System, Honda R&D Technical Review, April, 2010, p. 94-104
- (2) Liepa, P.: Filling Holes in Meshes, Proc. Eurographics/ACM SIGGRAPH Symp. Geometry Processing, p. 200-205 (2003)
- (3) Botsch, M., Kobbelt, L.: A Remeshing Approach to Multiresolution Modeling, Proc. Eurographics/ACM SIGGRAPH Symp. Geometry Processing, p. 189-196 (2004)
- (4) Edward, M. G., Stephan, J. W.: Object-oriented software for quadratic programming, ACM Transactions on Mathematical Software, Vol. 29, Issue 1, P. 58-81 (2003)
- (5) Gerald, E. F.: Curves and Surfaces for CAGD, The Frenet Frame, p. 191-194 (1993)
- (6) Shimizu, Y.: The Technique Generating Curves with Monotonous Curvatures for Aesthetic Surface Creation, UNISYS TECHNOLOGY REVIEW, Vol. 21, No. 3 (No. 71), p. 85-104 (2001)
- (7) Shimizu, Y., Sekido, K.: The Development of the Mesh Surface Generation G1-Connecting to Neighboring Surface, UNISYS TECHNOLOGY REVIEW, Vol. 19, No. 4 (No. 64), p. 1-26 (2000)

■ Author ■



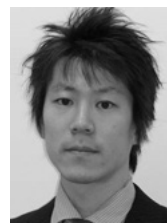
Tomoyoshi SHINOHARA



Keiko SUNOHARA



Saburo SAKAI



Hirotsugu KATSUBE



Daisuke IDE