High-Quality Sharp Features in Triangular Meshes

Masatake Higashi[1], Hiromu Inoue[2] and Tetsuo Oya[3]

[1]Toyota Technological Institute, higashi@toyota-ti.ac.jp
[2]Toyota Technological Institute, sd01009@toyota-ti.ac.jp
[3]Toyota Technological Institute, oya@toyota-ti.ac.jp

ABSTRACT

We propose a method which generates high-quality sharp features on triangular mesh surfaces. Triangular meshes, which are interpolated as subdivision surfaces, have been used to express compound surfaces, because they automatically satisfy continuity among arbitrary meshes. But the quality of sharp features such as creases or corners at an arbitrary part of a mesh is not investigated enough. Hence, we introduce triangular spline with C^0 continuity by inserting extended meshes at specified edges to explicitly control the shape of sharp features and to represent them with triangular Bézier patches. Then we apply this technique to control tangential planes along the specified edges.

Keywords: triangular mesh, triangular spline, continuity, crease.

1. INTRODUCTION

To represent industrial products composed by free-form surfaces like automotive bodies and electrical appliances, tensor product surfaces such as non-uniform rational B-spline surfaces and Bézier patches have been widely used. They can represent high-quality surfaces, but to connect them each other with continuity is rather difficult. Hence subdivision surfaces for arbitrary meshes [8] are getting to be used in the field of computer graphics because of their availability to any type of shape and automatic satisfaction of continuity. However, it has not been used for shapes of industrial products because they lack in class-A smoothness and controllability of surfaces.

There have been many studies on generating sharp features on a triangular mesh. Hoppe et al. [7] proposed a method to automatically determine the topological type of the surface, and the presence and location of sharp features. A key ingredient of their method is a new subdivision surface scheme that allows the modeling of surface features such as corners, boundaries, creases, and darts. Patterns of sharp features are shown in Fig. 1, where central 6 triangles are picked up and bold lines express sharp features. DeRose et al. [5] introduced a practical technique for constructing soft creases, that are variable-radius fillets and blends. Next, Biermann et al. [1] modified the subdivision rules for boundary vertices and to change normals at vertices to prescribed ones, and they also introduced a method for local sharp feature generation [2]. Then, Ying et al. [13] extended this normal control to nonmanifold surfaces which share

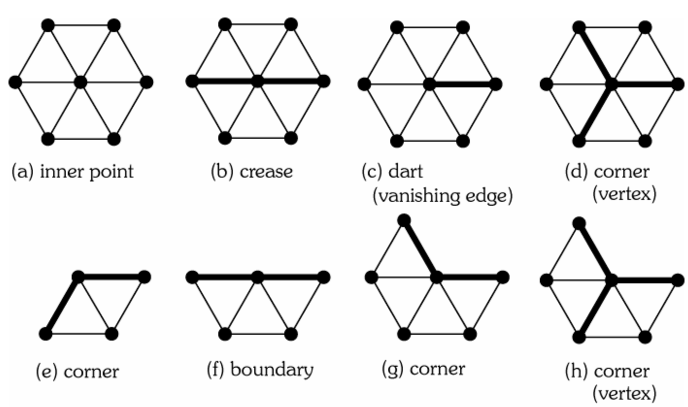


Fig. 1: Patterns of sharp features (a bold line is a sharp feature or a boundary).

the common edge and tangential plane. On the other hand, Sederberg et al. [11] presented non-uniform recursive subdivision surfaces which change the intervals of knots to control sharp features, and Ren et al. [10] proposed a method which changes weights of vertices for subdivision surfaces, while Vanraes and Bultheel [12] introduced degenerated control triangles.

The surfaces with sharp features obtained by above methods are not perfectly smooth for the use of stamping dies to manufacture products. Therefore, we study on generating and controlling high-quality sharp features. We introduce triangular spline with C^0 continuity by inserting extended meshes at specified edges to explicitly control the shape of sharp features instead of changing coefficients for subdivision or knot intervals, and represent them with triangular Bézier patches.

In this paper, we describe triangular spline and its representation by Bézier patches in Section 2. Next, we introduce a method to generate and control creases at specified edges along with a dart, that is a vanishing edge, in Section 3. Then in Section 4, we present a method to generate corners, that are intersections of creases, as well as boundary edges. In Section 5, we extend these methods to control tangential directions along the specified edges, and conclude the paper after discussion.

2. TRIANGULAR SPLINE AND BÉZIER PATCH

A triangular spline, which represents a surface for regular triangular meshes, is extension of a univariate B-spline to that for trivariate *u*, *v* and *w* barycentric coordinates [3], and a triangular spline surface is represented with Bézier patches. Here, a regular mesh satisfies that it is composed by vertices where six triangles are incident to except on the boundaries. The relation between control points for a B-spline surface and Bézier patches are deduced by Boem [4]. The relation is shown in Fig. 2.

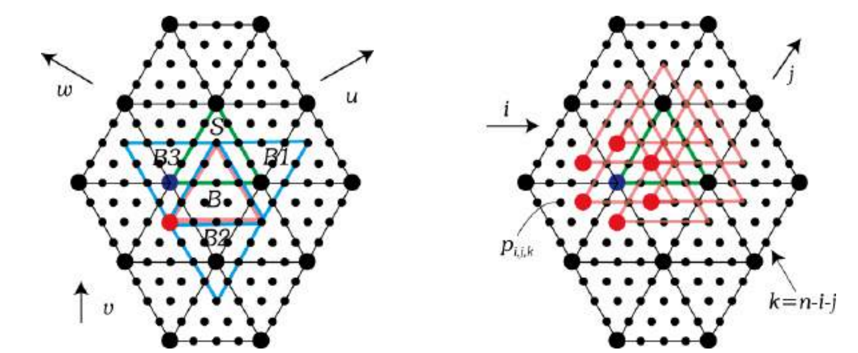
First, we generate spline sub-points dividing edges of spline triangles into four, connecting them by lines parallel to edges and generating intersection points as small dots shown in Fig. 2(a). When we generate quartic Bézier patches, whose vertices are 0.5 or -0.5 sifted along a coordinate: *u*, *v* or *w*, against those of the B-spline triangle. The central pink Bézier patch *B* in Fig. 2(a), which is shifted by -0.5 along *v*, is C^2 continuous to adjacent three blue Bézier patches *B1*, *B2* and *B3* which are shifted similarly from adjacent patches of the B-spline patch *S*. Now, we generate a Bézier patch corresponding to a green B-spline triangle in Fig. 2(b). We generate six Bézier patches, shown with pink lines, around the green one shifted by ±0.5 along one coordinate, and taking average of corresponding control points of Bézier patches we obtain a Bézier patch. A blue left lower vertex of the green Bézier patch is an average of red six vertices. This calculation is applied to 15 Bézier control points, then the patch shape is determined.

Let vertices of spine triangles be P_i $i=1, \ldots, n$, and let divided sub-points and corresponding Bézier control points (B-points) be $\boldsymbol{p}_{i,j,k}$ and $\boldsymbol{q}_{i,j,k}$. Then we get equations of B-points:

$$\boldsymbol{q}_{i,j,k} = \frac{1}{6}(\mathrm{EF}+\mathrm{E}^{-1}\mathrm{F}^{2}+\mathrm{E}^{-2}\mathrm{F}^{1}+\mathrm{E}^{-1}\mathrm{F}^{-1}+\mathrm{EF}^{-2}+\mathrm{E}^{2}\mathrm{F}^{-1})\boldsymbol{p}_{i,j,k} \quad (2.1)$$

Here, $\mathrm{E}\,\boldsymbol{p}_{i,j,k} = \boldsymbol{p}_{i+1,j,k}$, $\mathrm{F}\,\boldsymbol{p}_{i,j,k} = \boldsymbol{p}_{i,j+1,k}$, $\mathrm{E}^{-1}\,\boldsymbol{p}_{i,j,k} = \boldsymbol{p}_{i-1,j,k}$

E and F are shift operators for indices of sub-points by 1 along the triangular edge: *i* and *j*. The sum of indices for the sub-point of degree *n* satisfies $i+j+k=n$. Since parameters *u, v, w* are varied along the direction perpendicular to the edge of the triangle, the sift of sub-point by 0.5 along the coordinate is represented by the sift of indices of the sub-point as in Eqn. (2.1). An



(a) Bézier triangle and adjacent C^2 patches (b) Bézier point from B-spline sub-points
Fig. 2: Relation between B-spline and Bézier triangles.

equation of a n-degree Bézier patch is as follows [6].

$$\boldsymbol{S}(u,v,w)=\sum_{i,j,k=0}^{n}\frac{n!}{i!\,j!\,k!}u^i v^j w^k \boldsymbol{q}_{i,j,k} \qquad (2.2)$$

$$u+v+w=1, \quad i+j+k=n$$

Fig. 3 shows an example of a Bézier patch. A cubic Bézier control polygon is drawn with red bold lines along with its control points.

Fig. 4 shows an example of a B-spline mesh and generated Bézier patches from it. Fig. 4(a) shows a B-spline mesh, Fig. 4(b) Bézier patches and Fig. 4(c) a shaded picture drawn using Bézier patches.

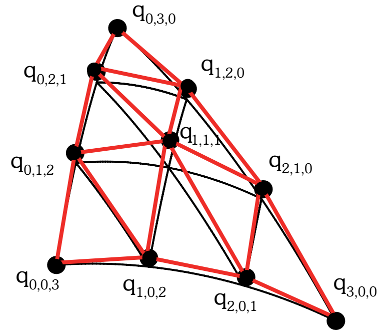
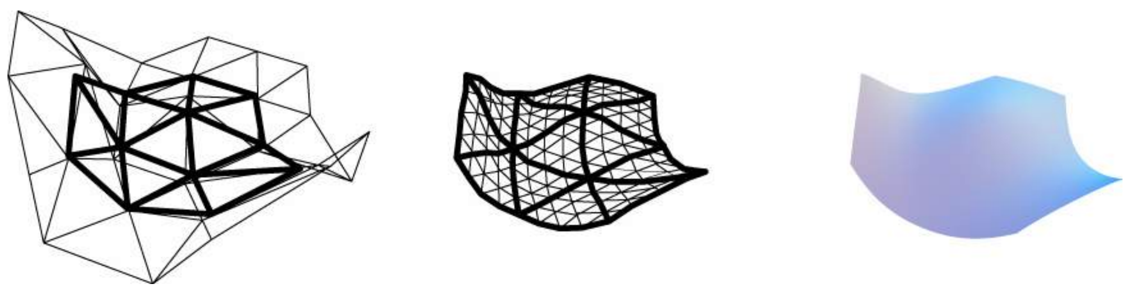


Fig. 3: Bézier patch and control points.



(a) B-spline and Bézier (bold lines) triangles (b) Quartic Bézier patches (c) A shaded picture of generated surface

Fig. 4: B-spline and Bézier triangles and generated surface.

3. CONTROL OF CREASE SHAPE AND VANISHING EDGE

In this section, we describe rules of subdivision for a mesh with sharp features, and introduce a method to express a crease for specified edges by utilizing extended spline meshes along them. Loop [8] introduced a subdivision surface which is extension of a regular spline mesh to a general triangular mesh. He showed rules of subdividing each triangle of a mesh into 4 triangles. Newly generated vertices are classified into vertex points and edge points according to their positions. Their coordinate values for vertex points and edge points are

$$\begin{aligned}\boldsymbol{v}^{r+1} &= \tfrac{1}{\alpha(n)+n}(\alpha(n)\boldsymbol{v}^r+\boldsymbol{v}_1^r+\cdots+\boldsymbol{v}_n^r),\\ &\alpha(n)=\tfrac{n(1-a(n))}{a(n)}, \quad a(n)=\tfrac{5}{8}-\tfrac{1}{64}(3+2\cos(2\pi/n))^2,\\ \boldsymbol{v}_i^{r+1} &= \tfrac{1}{8}(3\boldsymbol{v}^r+3\boldsymbol{v}_i^r+\boldsymbol{v}_{i-1}^r+\boldsymbol{v}_{i+1}^r).\end{aligned} \qquad (3.1)$$

Here, n is a valence of a vertex, which is a number of incident edges, and $\boldsymbol{v}^r$ is a concerned vertex at the r-th subdivision and $\boldsymbol{v}_i^r$ is its neighbor vertices. Then a crease is expressed by Hoppe et al. [7] and modified by Biermann et al. [1]. A crease vertex and a corner vertex are calculated as

$$\begin{aligned}\boldsymbol{v}_i^{r+1} &= \tfrac{1}{2}(\boldsymbol{v}_i^r+\boldsymbol{v}_{i-1}^r), && \text{: a crease vertex}\\ \boldsymbol{v}^{r+1} &= \boldsymbol{v}^r, && \text{: a corner vertex}\\ \boldsymbol{v}_i^{r+1} &= \tfrac{1}{8}(3\boldsymbol{v}_i^r+5\boldsymbol{v}_{i-1}^r). && \text{: a non - regular crease vertex}\end{aligned} \qquad (3.2)$$

Vertex and edge subdivision masks are shown in Fig. 5. From Fig. 5 (a) to (c) are vertex subdivisions and from (d) to (e) are for edge subdivisions. Fig. 5 (g) shows a subdivision rule for a non-regular crease edge modified by Biermann et al. which changes a mask according to a number of adjacent polygons, k.

$$\boldsymbol{v}_i^{r+1} = \tfrac{1}{8}((6-8\gamma)\boldsymbol{v}^r + 8\gamma\boldsymbol{v}_i^r + \boldsymbol{v}_{i-1}^r + \boldsymbol{v}_{i+1}^r) \quad \text{: a modified non-regular crease vertex} \tag{3.3}$$

$$\gamma = \tfrac{1}{2} - \tfrac{\cos\theta}{4}, \quad \theta = 2\pi/k \text{ (dart vertex)}, \quad \theta = \pi/k \text{ (crease vertex)}$$

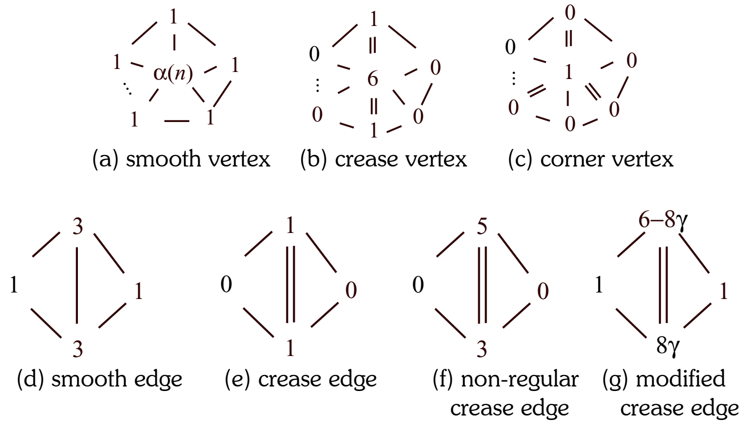


Fig. 5: Vertex subdivision masks.

Now, we introduce sharp features in a spline mesh. Along the specified edges, we separate a mesh and generate triangles at the opposite sides (see Fig. 6). If we generate new triangles so as to make parallelograms with the original ones, a generated sharp edge should be represented only by the specified spline edges. This is because the control points for the generated edge which is shared by the both-side surfaces are represented without influence of inner sub-points of the triangles. In Fig. 6, pairs of green and red triangles are parallelograms. The position of new vertex P_k' is

$$\boldsymbol{P}_k' = \boldsymbol{P}_j + \boldsymbol{P}_i - \boldsymbol{P}_k. \tag{3.4}$$

Bézier patches are similarly calculated for both surfaces from spline triangles including extended ones. The curve for the specified edges is coincide with that for a cubic spline. This is the same to the case of a subdivision surface.

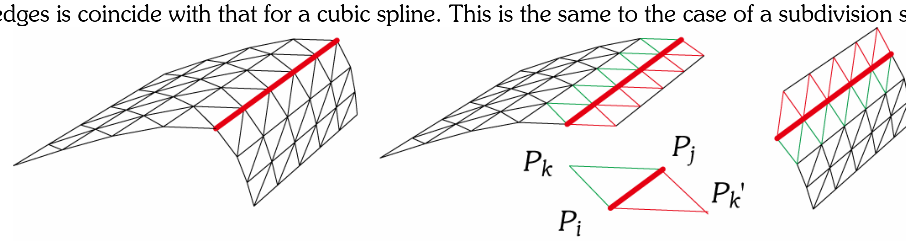


Fig. 6: Representation of a crease edge by B-spline triangles.

Next, we consider a dart edge which is a vertex where a crease disappears. A dart in a subdivision surface is expressed in Eqns. (3.1) and (3.2), but smoothness at the vicinity of the vertex is not enough. Its shape is like a dart of a dress, which is not desirable for that on an automotive panel. Connecting extended and original triangles as shown in Fig. 7, we can generate a surface to vary smoothly according to the extended triangles. The sharp edges is vanishing between P_i and P_j, and the surface is changing very smoothly between them according to the transit triangles: $P_iq_iq_j$ and $P_iq_i'q_j'$. At P_j, continuity between surfaces is C^2.

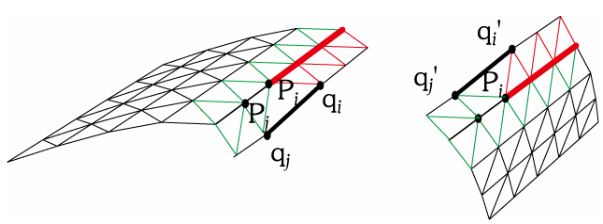


Fig. 7: Generation of a dart (vanishing edge).

Lastly, we describe a method to change sharpness at the specified edge. The shape of the sharp features generated using Eqn. (3.4) is similar to that by a subdivision surface. If we change the positions of the vertices of the extended triangles as shown in Fig. 8, we can control the sharpness along the edge. Let P_M and P_N be midpoints of P_i and P_j and P_k and P_l, and let *s* and P_G be a ratio dividing a distance between P_N and P_M and a divided point. When we determine extended vertices Q_k and Q_l by extending P_k and P_l from P_G by the same distance, the sharp edge is to be interpolated only by the control polygons of the edges similarly as Eqn. (3.4). An angle between both sides becomes wider as *s* decreases to 0, and the shape approaches the C^2 surface. The vertices of the extended triangles are expressed as follows.

$$\begin{aligned} \boldsymbol{Q}_k &= \boldsymbol{P}_k + 2(\boldsymbol{P}_G - \boldsymbol{P}_k) = s(-\boldsymbol{P}_k + \boldsymbol{P}_i + \boldsymbol{P}_j - \boldsymbol{P}_l) + \boldsymbol{P}_l \\ \boldsymbol{Q}_l &= \boldsymbol{P}_l + 2(\boldsymbol{P}_G - \boldsymbol{P}_l) = s(-\boldsymbol{P}_k + \boldsymbol{P}_i + \boldsymbol{P}_j - \boldsymbol{P}_l) + \boldsymbol{P}_k \end{aligned} \tag{3.5}$$

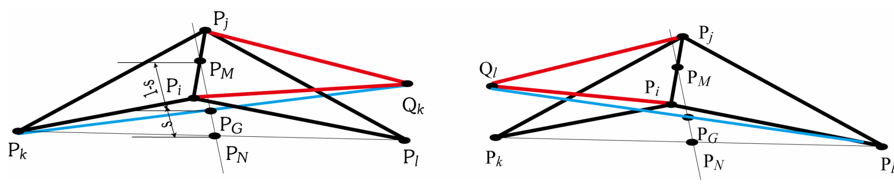


Fig. 8: Control of sharpness of the edge.

We show examples of sharp features in Fig. 9. Fig. 9(a) is a surface without specification of a crease, then it is C^2 everywhere. Fig. 9(b) is a surface generated according to the specification of a crease edge in Fig. 9(d), and Fig. 9(c) shows a changed sharpness of the crease edge by setting *s*=0.5. Fig. 9(d) is a surface with a crease edge which disappears at the middle according to the specification in Fig. 9(e). The surface changes smoothly from C^0 to C^2, which we call a vanishing edge and very desirable for a shape as an automotive body. Fig. 9(g) shows a dart edge by subdivision, where a sharp edge disappears quickly.

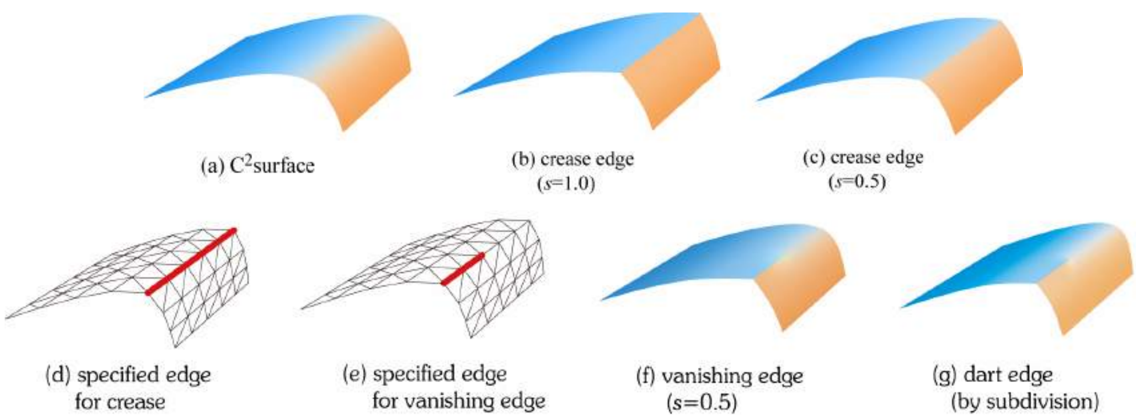
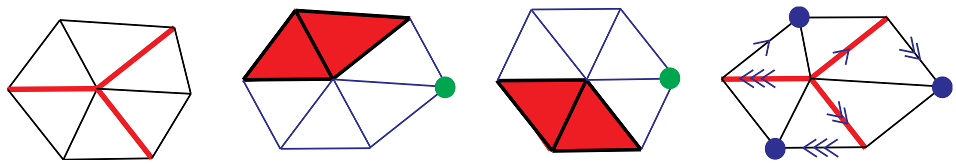


Fig. 9: Examples of crease edges.

4. CORNERS AND BOUNDARY EDGES

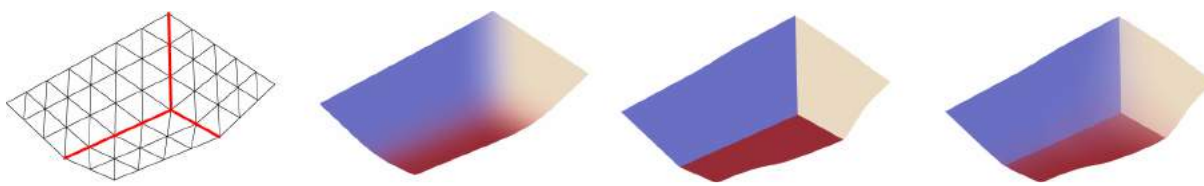
In this section, we present a method for treating corners which are intersection points of multiple crease edges. We can represent surface equations around corners by extending triangles similarly to crease edges. However, if we extend them for each sector at the corner, we obtain different vertices among sectors. Fig. 10 shows an example of three sectors coincident to a corner. The positions of the vertex calculated by the average in Eqn. (2.1) for Fig. 10(b) and Fig. 10(c) are different, because the positions of green points calculated for sector A and B are different. Hence if we want a consistent vertex position for a corner, we must change inner shapes of the sectors. There is freedom for the change, so we take a very simple way. We move the position of inner vertex in each sector so as to make a parallelogram from two triangles as shown if Fig. 10(d). Then we can get the same vertex of the corner for the three sectors.



(a) original data (b) extension of sector A (c) extension of sector B (d) modification of inner point

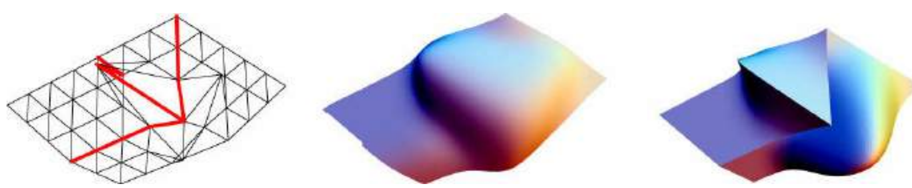
Fig. 10: Extension of triangles at a corner.

We can apply the method of changing sharpness in Section 3 to the corners. When we extend the triangles on the same plane, the obtained vertex is coincident to the original control point. Therefore, the obtained corner vertex is different from other points on the boundary edge which are interpolated by a cubic B-spline, and the curvature at the corner vertex is zero. Hence we control the curve shape by using the ratio in Eqn. (3.5). Fig. 11 shows examples of surfaces with corners; Fig. 11(b) is a shape without a corner vertex, and Fig. 11(c) and (d) are shapes with a corner vertex specified Fig. 11(a). Fig. 11(d) has more natural shape at the vertex by setting s=0.5 than Fig. 11(c). There are several patterns of sectors at the corner as for a regular mesh. Hence we have to set up rules corresponding to each pattern. But the basic rule is the same: If the extended vertices are not coincident, then we change the positions of internal vertices. An example is shown in Fig. 12, where two sectors at the corner have one triangle respectively and the other one has four triangles.



(a) input mesh and specified edges (b) shape without corner (c) shape with sharp corner (d) shape with natural corner

Fig. 11: Generated corner.



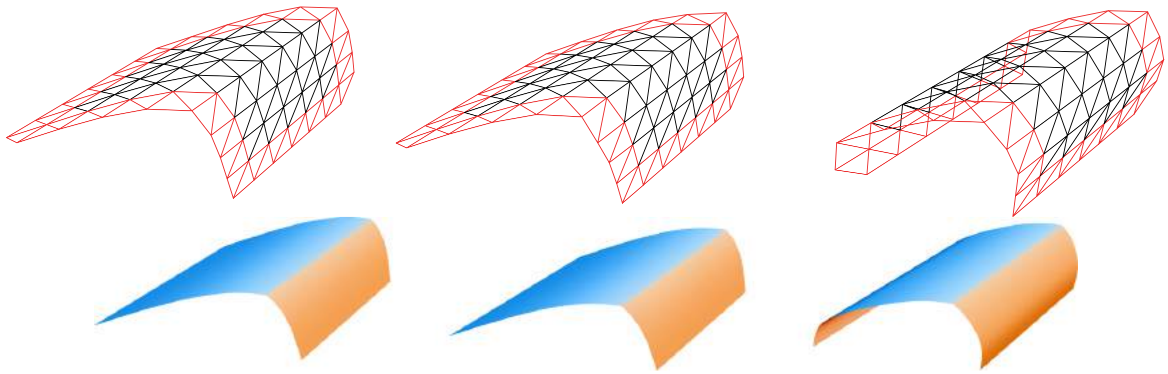
(a) input mesh and specified edges b) shape without corner (c) shape with sharp corner

Fig. 12: Corner with different sectors.

Next, we discuss about boundaries along with those with corners. A surface along a boundary is a special case of the crease. It is determined without the opposite surface of the crease. Hence to control the position of the boundary, we have freedom, so we extend triangles by two ways: One is that extended triangles are on the same plane of the boundary triangles and the other is that they are extended as smoothly as the vertex of the extended triangle is on a parabola interpolated by the vertices inside the boundary. For the corner vertices shown in Fig. 1(e) and (g), we can extend the triangles without restriction of the other sector, but for the corner vertex in Fig. 1(h), we have to treat it similarly as the intersection vertex described above. Fig. 13 shows examples of boundaries with different extension methods. Red triangles are extended to make boundaries. Fig. 13(a) and (b) do not show much difference. Fig. 13(c) is generated for rounding the both ends of the surface according to the modification of tangential planes.

5. CONTROL OF TANGENTIAL PLANE ALONG EDGE

We extend the concept of controlling sharp features by extension of triangles to controlling tangential planes of a mesh. We specify tangential planes along the specified edges by changing the positions of the vertices of the adjacent triangles to the edges, while we only extend triangles for generating sharp features. This is determined in the subdivision



(a) extension on the same plane (b) extension by parabola interpolation (c) change of tangential planes

Fig. 13: Boundary with different extension methods.

surfaces by inserting subdivided vertices to approach a point on the tangential plane [1], but instead we directly specify the B-spline triangles. Fig. 14 shows triangles for specifying tangential planes. We can generate surfaces with C^2 continuity along the edges as follows. We move vertex P_l of a triangle on one side to P_l' so that it is on the specified tangential plane and extend it to the other side through P_G which is a divided point between P_M and P_N similarly in Fig. 8. Then we also move a vertex on the other side to P_k' and generate an extended vertex Q_k similarly. Red triangles in Fig. 14 are used for specifying a tangential plane; Solid lines represent modified triangles and dashed lines show extended triangles. All the vertices of four triangles are arranged on the same blue line, which realizes C^2 continuity along the edges.

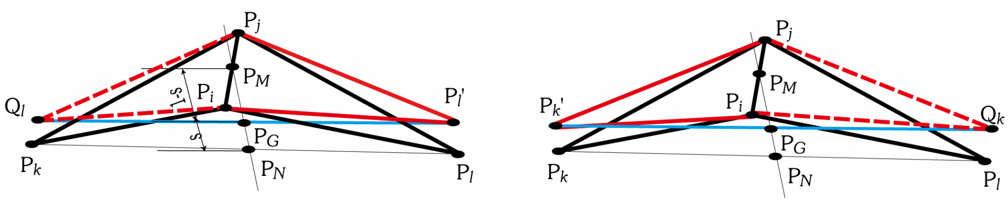
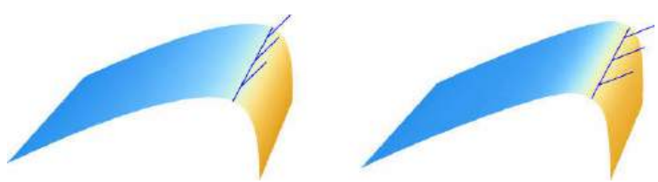


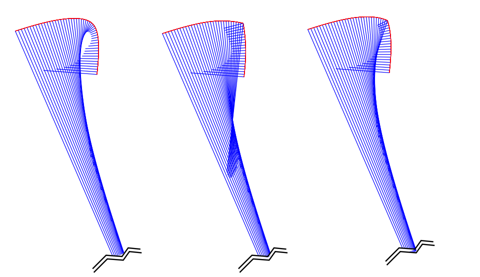
Fig. 14: Control of tangential plane along edges.

Fig.15 shows an example of a shape by the tangential control compared to the original one. We use the similar mesh with Fig. 9, and change the tangential directions. The curvature is getting larger by the change of the normal vectors of tangential planes.



(a) original shape (b) modified shape by tangential plane

Fig. 15: Modification of a shape by changing tangential plane.



(a) without crease (b) sharp feature (s=1.0) (c) sharp feature (s=0.5)

Fig. 16: Curvature profiles of parametric lines crossing a crease.

6. DISCUSSION

In this section, we discuss the quality of the generated surfaces by the sharp feature control. For subdivision surfaces, they divide a mesh into that consisting of smaller triangles according to the rules. Hence, the quality depends on the original mesh and the limit surface of the subdivision. In our approach, we define the sharp features directly by extended triangles, so we can control the shape. We show quality of the generated surfaces by displaying curvature variation along the parametric lines. Fig. 16 shows curvature profiles for parametric lines crossing a sharp feature. Here, a curvature profile illustrates

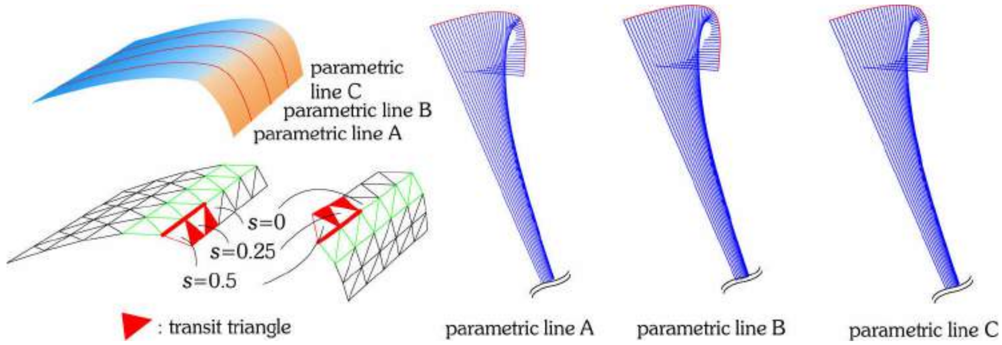


Fig. 17: Curvature profiles for a vanishing edge.

normal lines with 1/10 radii at points along the specified curve, so it expresses the variation of curvature visually. Fig. 16(a) is a shape without a sharp feature, (b) with a crease edge by s=1.0, and (c) with a crease edge by s=0.5 corresponding to figures in Fig. 9. We get a better profile for (c), which is composed by monotone curvature variation, than that for (b) which corresponds to a subdivision surface. Next, in Fig. 17, we show the change of a surface from with an edge to without one when coefficient s is changed gradually: 0.5, 0.25 and 0. The curvature profiles of three parametric lines which cross a vanishing edge are very smooth because the shapes of transit triangles change smoothly.

7. CONCLUSION

We have introduced a method for generating sharp features in a B-spline triangle mesh as well as one for specifying tangent directions along the specified edges. By using these methods we have obtained surfaces with higher-quality than those by subdivision methods. We treat a regular mesh in this paper, but we can extend this method to that of a non-regular mesh by applying the method proposed by Peters [9]. This is our next target of the research. This research was partly supported by the High-Tech Research Center project from the Ministry of Education, Sports, Culture, Science and Technology, Japan.

8. REFERENCES

[1] Biermann, H.; Levin, A.; Zorin, D.: Piecewise Smooth Subdivision Surfaces with Normal Control, Proc. SIGGRAPH2000, 2000, 113-120.
[2] Biermann, H.; Boier-Martin, I.M.; Zorin, D.; Bernardini F: Sharp Features on Multiresolution Subdivision Surfaces, Graphical Models, 64(2), 2002, 61-77.
[3] Boem, W.: The De Boor Algorithm for Triangular splines, in: Surfaces in Computer Aided Geometric Design, North-Holland Publishing Company, 1983, 109-120.
[4] Boem, W.: Generating the Bezier Points of Triangular Spline, in: Surfaces in Computer Aided Geometric Design, North-Holland Publishing Company, 1983, 77-91.
[5] DeRose, T.; Kass, M.; Truong, T.: Subdivision Surfaces in Character Animation, Proc. SIGGRAPH98, 1998, 85-94.
[6] Farin, G.: Smooth Interpolation to Scattered 3D Data, in: Surfaces in Computer Aided Geometric Design, North-Holland Publishing Company, 1983, 43-63.
[7] Hoppe, H.; DeRose, T.; Duchamp, T.; Halstead, M.: Piecewise Smooth Surface Reconstruction, Proc. SIGGRAPH94, 1994, 295-302.
[8] Loop, C.: Smooth subdivision surfaces based on triangles. Master's Thesis, Department of Mathematics, University of Utah, 1987.
[9] Peters, J.: Smooth Patching of Refined Triangulations, ACM Transactions on Graphics, 20(1), 2001, 1-9.
[10] Ren, B.Y.; Li, X.D.; Wu, Z.Q.; Hagiwara, I.: Local Sharp Feature Generation and Shape Control of Recursive Subdivision Surface, JSME International Journal Series C, 2005, 170-175.
[11] Sederberg, T. W.; Sewell, D.; Sabin, M.: Non-Uniform Recursive Subdivision Surfaces, SIGGRAPH98, 1998, 387-394.
[12] Vanraes E.; Bultheel. A.: Modeling sharp features with tangent subdivision, In M. Daehlen, K. Mörken L. Schumaker (Eds), Mathematical methods for curves and surfaces, Nashboro Press, 2005, 362-372.
[13] Ying, L.; Zorin, D.: Nonmanifold Subdivision, Proc. IEEE Visualization, 2001, 325-332.