Vol. 41 No. 3 Transactions of Information Processing Society of Japan Mar. 2000

Regular Paper

Introducing Feature Values for Effective Specification of Polyhedral Networks

Zheng Xu† and Kunio Kondo†

The recursive subdivision of polyhedral networks, often called polyhedral subdivision, has become one of the basic tools in Computer Aided Geometric Design (CAGD) for modeling complex surfaces since the first two methods were proposed by Catmull and Clark (1978) and Doo and Sabin (1978). Unfortunately, it is still inconvenient to use polyhedral subdivision to generate surfaces which are the same as or close to the surfaces designers want to achieve over simple and rough polyhedral networks. The efficient and easy way of achieving this is to define precision polyhedral networks. However, it is obviously very troublesome for users to generate precise, complex polyhedral networks. In order to solve this problem, the concept of feature values for vertices, edges, and faces of a polyhedron is introduced in this paper, and the problem of making a subdivision process with the feature values of a polyhedron is studied. With these feature values, a polyhedron will be divided once before using a polyhedral subdivision method to generate surfaces. The newly generated polyhedrons can be controlled and modified easily by adjusting the feature values. This paper proposes an efficient way to create polyhedral networks for modeling complex surfaces.

1. Introduction

1.1 Background

As the polyhedral subdivision process provides a simple way to generate surfaces over polyhedral networks or irregular topologies, it is widely used in CAGD for modeling complex surfaces. The fundamental idea goes back to Chaikin's algorithm[1] which generates a quadratic B-spline curve from a polygon by successively cutting its corners. In 1978, the first two polyhedral subdivision methods were introduced by Catmull and Clark[2], and Doo and Sabin[3]. They applied Chaikin's idea to the generation of surfaces. In these methods, an initial polyhedral network is successively modified on its edges and corners. In the limit, a surface, often called the subdivision surface, can be generated over the polyhedral network. Because *no global parameterization is possible in general in these surfaces*[10], these polyhedral subdivision methods arouse the interest of researchers. Many approaches have been attempted to follow or extend the polyhedral subdivision methods. Nasri extended the Doo-Sabin method to generate a surface which interpolates some or all vertices of a polyhedron[4),6], and interpolates the vertices with normal conditions[5]. Moreover, he proposed an algorithm to generate subdivision surfaces which interpolate B-spline curves[7]. Peters[13] extended the subdivision technology to generate surfaces by separating singular regions after a few subdivision steps. Halstead, et al.[14] proposed an interpolation method using Catmull-Clark surfaces. More detailed discussions on subdivision surfaces can be found in Ref. 4)–9), 11)–15).

1.2 Motivation

In the polyhedral subdivision method, the initial polyhedral network decides the final shapes of surfaces. Thus, to model a complex surface, a complex or precise polyhedral network must be defined. **Figure 1** illustrates the derivation of a subdivision surface with a sharp corner and a distorted area from a simple cube (Fig. 1 (a)). In the Doo-Sabin subdivision method, another polyhedron, illustrated in Fig. 1 (b), must be defined, then the shape shown in Fig. 1 (c) can be obtained. Generally, a 3D model can be input by a 3D input device or created with an interactive editor interface. The designer usually uses 3D input devices, for example a 3D laser scanner, to generate 3D data of a shape from an physical model. Using polyhedral meshes to reconstruct shapes from the input data is a primitive way. But the number of polyhedral meshes is too huge to manipulate. On the other hand, in many cases, physical models are unavailable and designers must create them by hand. In this case, a basic method is using surface patches to represent surfaces, but it is very expensive to deal with complex surfaces. A polyhedral subdivi-

† Department of Information and Computer Sciences, Saitama University

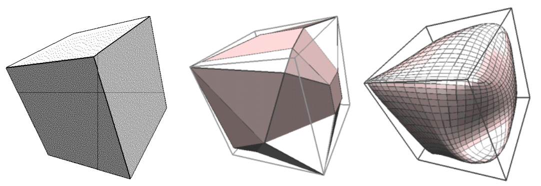


Fig. 1 Generating a surface with a sharp corner and a distorted area from a cube. (a) A cube. (b) The cube and a new polyhedron. (c) The cube and the shape generated from the new polyhedron in (b).

sion method is a good choice. However, in the subdivision method, it is difficult to predict the final shape from the original polyhedral networks precisely, and generating complex polyhedral networks is difficult work for designers. Although simple polyhedra can be generated easily, it is difficult to develop surfaces from simple and rough polyhedral networks that are the same as, or close to, the surfaces designers wish to model.

Many approaches to the subdivision of surfaces have previously been proposed. Some interpolation methods were proposed by Nasri [4),6),7)], Halstead, et al. [14)] and Dyn, et al. [12)]. A set of local shape handling methods for controlling the quality of the final shape was proposed by Brunet [9)]. However, basically these methods focused on how to control and modify surfaces generated from a polygon that had already been selected, and did not solve the problem of how to model a complex polyhedral network efficiently for controlling and generating subdivision surfaces. To solve this problem, the concept of feature values for vertices, edges, and faces of a polyhedron is introduced in this paper.

The feature values can be considered as parameters of a polyhedron. With these feature values, a simple and rough polyhedron is divided once before using the subdivision method, then a complex and precise polyhedral network can be derived easily. Designers can control the shape of new generated polyhedral networks easily by giving and adjusting feature values and a complex and precise polyhedral network with some shape features can be developed easily from a simple and rough polyhedral network. For example, in Fig. 1, we can generate a new complex polyhedron Fig. 1 (b) easily by assigning feature values to the vertices and edges of the cube. Then, the final shape generated by the Doo-Sabin method in Fig. 1 (c) can easily be obtained. In this paper, we use the Doo-Sabin subdivision method to generate surfaces. Users can extend our method and use other subdivision approaches to model final shapes.

As our approach is based on Doo-Sabin subdivision surfaces and the Nasri interpolation method, they will both be described in detail in the following section. In Section 3, we introduce the definition of feature values and the subdivision process using the feature values. Section 4 presents three examples to illustrate the effect of subdivision with feature values. Finally, we come to conclusions and propose future work.

2. Doo-Sabin Subdivision Surfaces and the Nasri Interpolation Method

In the Doo-Sabin method, surfaces are generated from polyhedral networks by successively cutting the corners and edges of the polyhedron. The Doo-Sabin algorithm can be described as follows and is illustrated in **Fig. 2**. Some terms proposed by Nasri [4)] are used hereunder:

(1) For every vertex V_i of the polyhedron P_i, a new vertex V_i', called an *image*, is generated on each face adjacent to V_i.
(2) For each face F_i of P_i, a new face, called an *F-face*, is made by connecting the *images*, the vertices V_i' generated in Step 1.
(3) For each edge E_i common to two faces F_i and F_i', a new four-sided face, called an *E-face*, is made by connecting the *images* of the end vertices of E_i on the faces F_i and F_i'.
(4) For each vertex V_i, where n faces meet, a new face, called a *V-face*, is made by connecting the *images* of V_i on the faces meeting at V_i.

The *image* vertices V_i' generated in Step 1 are functions only of the vertices of P_i. That is:

$$V_i' = \sum_{j=1}^{n} a_{ij} V_j$$

where V_j are the vertices of the old faces, V_i' is the new vertex of V_j, and a_{ij} are weights.

$$a_{ij} = \frac{n+5}{4n} \qquad \text{for } i = j,$$

$$a_{ij} = \frac{3 + 2\cos(2\pi(\frac{i-j}{n}))}{4n} \qquad \text{for } i \neq j.$$

Figure 2 (a) illustrates the *F-face*, *E-face*, and *V-face* of the Doo-Sabin subdivision method. A face of a polyhedron is enclosed in solid nodes and the generated *F-face*, *E-face* and *V-face* are

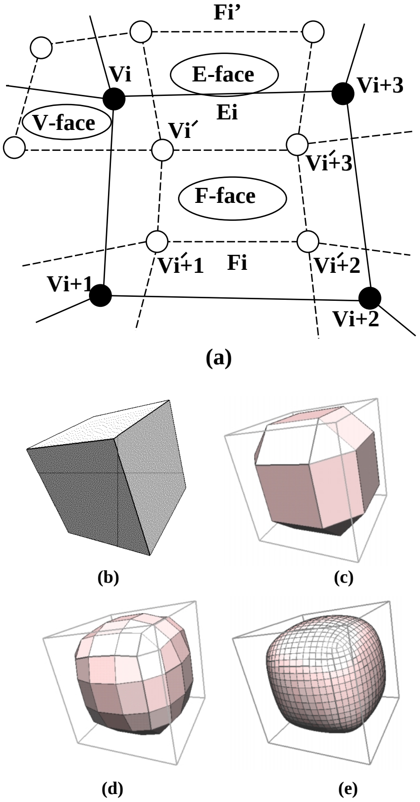


Fig. 2 The Doo-Sabin subdivision method. (a) Three types of faces. (b) A cube. (c) The cube and its first subdivision. (d) The cube and its second subdivision. (e) The cube and its fourth subdivision.

enclosed in hollow nodes. Figure 2 (b)–(e) illustrate a cube and the Doo-Sabin subdivision process over it.

Nasri [4),6)] extended the Doo-Sabin method by generating a surface which interpolates some or all vertices of a polyhedron. **Figure 3** illustrates an interpolation procedure. In Nasri's approach, a new polyhedron with a set of vertices $W = (W_i)_{1 \leq i \leq n}$, having the same topology as the set $V = (V_i)_{1 \leq i \leq n}$ for the edges and the faces of the initial polyhedron, is first obtained. There is a *one-to-one* mapping between the vertices of the sets V and W, such that W_k corresponds to V_k. Furthermore, the new polyhedron is such that each flagged vertex in V is the centroid of the *V-face* generated from its corresponding vertex in W. There is a simple linear relation between a V_i and the elements of the set W:

$$V_i = \frac{1}{m} \sum_{k=1}^{m} W_{ik}$$

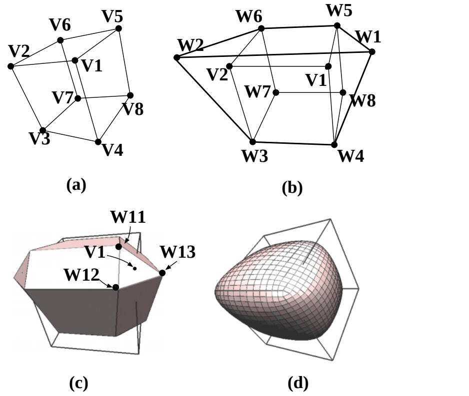


Fig. 3 Interpolation of two points of a cube using Nasri's method. (a) A cube. (b) The cube and the new polyhedron (constructed by linking $W_1, W_2, \ldots, W_8$) for interpolating two points. (c) The cube and its first subdivision and the images (W_{11}, W_{12}, W_{13}) corresponding to V_1. (d) The cube and its fourth subdivision.

where m is the number of faces meeting W_i (or V_i) and W_{ik} are the vertices of the *V-face* generated from W_i by the Doo-Sabin subdivision process. Thus, we can generate a subdivision face that interpolates some vertices of the initial polyhedron.

3. Subdividing Polyhedra Using Feature Values

3.1 The Definition of Feature Values

A polyhedron is composed of faces, edges, and vertices, so feature values are also defined for faces, edges, and vertices of a polyhedron. Whether to keep a face of a polyhedron as a plane in the subdivision process is defined as a *face feature value*, or *FFV*. If the *FFV* of a face is equal to 0, this means the face will be a plane and is called a *plane face*. If the *FFV* of a face is equal to 1, this means the face will be processed into a surface and is called a *surface face*.

The round extent along an edge of a polyhedron is defined as an *edge feature value*, or *EFV*. Each edge has different *EFV*s for the faces the edge is common to.

The sharp degree of a vertex of a polyhedron is defined as a vertex feature value, or *VFV*. Each vertex has different *VFV*s for the faces the vertex is common to.

With these feature values, a polyhedron is divided into a new polyhedron, and then the Doo-Sabin subdivision process is carried out over the new polyhedron. **Figure 4** illustrates a polyhedron and a generated surface with some feature

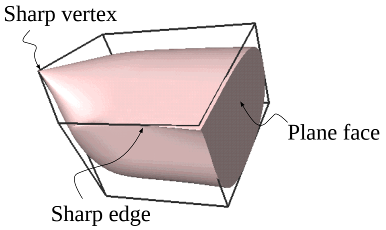


Fig. 4 A polyhedron and a generated surface with some feature values.

values.

3.2 The Subdivision Process with Feature Values

The subdivision process with feature values is carried out in different ways according to the type (*surface face* or *plane face*) of a face of a polyhedron.

3.2.1 Surface Face

When a face's FFV is equal to 1, this means the face is a *surface face*. The subdivision process is shown in **Fig. 5** and described as follows:

(1) For each n-sided *surface face* S_i, let the point C_i be the centroid of S_i.

(2) For each edge E_i of S_i, let E_i' be a line parallel to E_i, and let EFV_i, the feature value of the edge E_i on S_i, be the ratio of the distance between edge E_i and line E_i' to the distance between edge E_i and point C_i. Obviously, the value of EFV_i on S_i is a real number between 0.0 and 1.0 and the round extent on an edge will become larger as the EFV becomes larger.

(3) For each face S_i, let $\overline{V_i}$ (the hollow triangle nodes in Fig. 5) be the cross point of two contiguous lines generated in Step 2 (in Fig. 5, corresponding to $\overline{V_i}$, the two lines are E_i' and E_{i+4}'). According to $\overline{V_i}$ and the corresponding vertex V_i of S_i, let V_i' (the hollow circle nodes in Fig. 5) divide the line segment V_i, $\overline{V_i}$, and the ratio of the distance between V_i and V_i' to the distance between V_i and $\overline{V_i}$ is defined as VFV_i, the feature value of vertex V_i on the face S_i. Obviously, the value of VFV_i is a real number between 0.0 and 1.0 and a vertex will become sharper as the EFV becomes smaller. The new vertex V_i' is called an *image vertex*, as in the Doo-Sabin method. An *F-face* can be constructed by linking all *image vertices* (hollow circle nodes in Fig. 5) of S_i.

(4) If there are no *plane faces* defined in the polyhedron, we go to the next step. If there are *plane faces* defined in the polyhedron, we make a subdivision process on *plane faces*.

(5) The procedures for generating new *E-faces* from each edge and *V-faces* from each vertex are similar to the processes of the Doo-Sabin method.

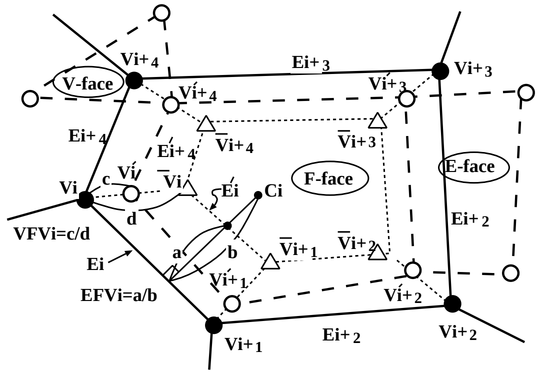


Fig. 5 The subdivision process with feature values on a *surface face* and its three types of faces enclosed in hollow circle nodes.

3.2.2 Plane Face

When the FFV of a face is 0, the face is treated as a *plane face*. The subdivision process will be carried out after finding all *F-faces* of *surface faces*. The *image vertices* (hollow circle nodes in Fig. 5) on *surface faces* surrounding a *plane face* will be used to construct an *F-face* of a *plane face*. The EFVs and VFVs on the *plane face* are ineffective. The generated *F-face* will be treated as a plane. The subdivision process is shown in **Fig. 6** and described as follows:

(1) For each edge E_i of a *plane face* P_i, test the FFV of the other face sharing the edge E_i with face P_i. If the FFV is 1, this means the face is a *surface face*, flagged in S_i. Suppose V_i' and V_{i+1}' (hollow nodes in Fig. 6 (a)) are the *image vertices* of the end of E_i on the face S_i, and let V_{si} and V_{ei} be the feet of the perpendiculars from V_i' and V_{i+1}' to the edge E_i, respectively.

If the FFV of another face sharing the edge E_i is 0, this means the face is a *plane face*. The end points of the edge E_i will be taken as V_{si} and V_{ei}, respectively (see edge E_{i+1} in Fig. 6 (a)). All vertices V_{si} and V_{ei} corresponding to each edge of P_i are referred to as *image vertices* (solid square nodes in Fig. 6 (a)) of the *plane face*.

(2) For each face P_i, linking the *image vertices* (solid square nodes in Fig. 6 (a)) to replace the vertices $V_i, V_{i+1}, \ldots, V_{i+m}$ (solid circle nodes in Fig. 6 (a)) of P_i, we construct an *F-face* P_i' of P_i.

(3) For *F-face* P_i' of P_i, there are several ways to ensure it remains a plane during the subdivision process. One method is to dou-

ble the topology data of P_i'. However, when two *plane faces* are connected, a correct result cannot be reached. Here, we use the interpolation method proposed by Nasri [4),6)]. All *E-faces* and *V-faces* surfaces connected with P_i' will be treated as open polygons.

To interpolate all edges of P_i', we construct a new *F-face* P_i'' with a set of vertices V_{si}', V_{ei}' (hollow triangle nodes in Fig. 6 (b)) as the reflection of the set of V_i', V_{i+1}' (hollow circle nodes in Fig. 6 (b)) about each edge V_{si}, V_{ei} (solid square nodes in Fig. 6 (b)) of P_i', respectively. The special case is the generation of new end vertices of the edge that two *plane faces* share. The vertex is called a *corner vertex*. Referring to Fig. 6 (b), two *corner vertices* are flagged as solid triangle nodes. Because the vertex C, the vertex of the original polyhedron, is the centroid of the polygon linked by p_1, p_2, p_3, and p_4, and we know the values of p_1, p_2, p_3, and C, the *corner vertex* can be obtained by:

$$p_4 = 4C - p_1 - p_2 - p_3.$$

More detailed discussion of this interpolation method can be found in Nasri's paper [4),6)]. The new *F-face* P_i'' can be obtained by linking all triangle nodes in Fig. 6 (b).

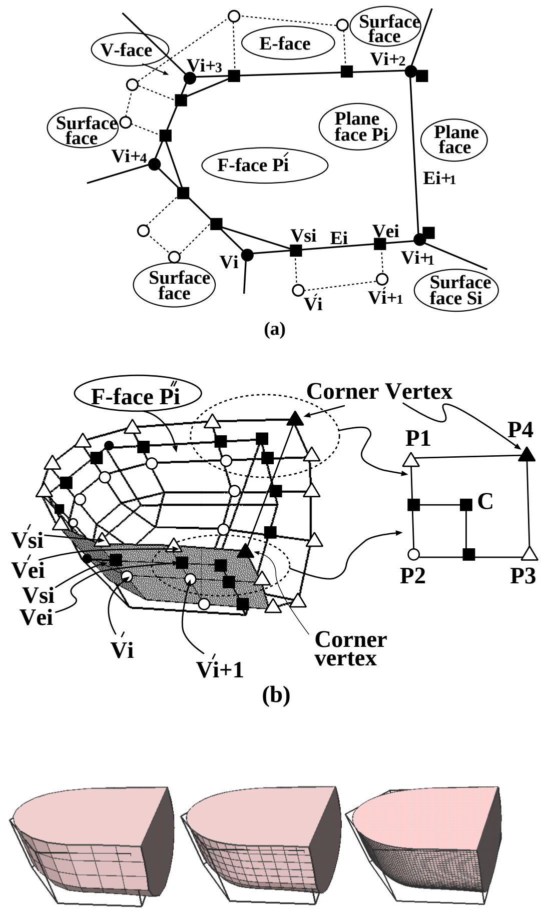


Fig. 6 The subdivision process using feature values on a *plane face*. (a) A *plane face* and its *F-face*, *E-face*, and *V-face*. (b) Reconstruction of three types of faces for interpolating all edges of the old *F-face* P_i' generated in (a). The result is a new *F-face* P_i'' (linked by triangle nodes). (c) The first Doo-Sabin subdivision over the reconstructed polyhedron. (d) The second Doo-Sabin subdivision process. (e) The fourth Doo-Sabin subdivision process.

After constructing the new *F-face* P_i'' of P_i, we go to step 5 in Section 3.2.1 to compute the *E-faces* and *V-faces*.

3.2.3 The Special Case of Subdividing a Polyhedron with Feature Values

In some cases, we will get a wrong sequence of vertices in the subdivision process. For example, in **Fig. 7**, the face is a *surface face* and is surrounded by *surface faces*. If the EFV_1 and EFV_4 are near 1, and EFV_5 is near 0, then we will get an incorrect sequence of vertices $\overline{V_1}$, $\overline{V_2}$, $\overline{V_3}$, $\overline{V_4}$, $\overline{V_5}$. To construct a correct type *F-face*, we must modify the sequence by changing the order of $\overline{V_4}$ and $\overline{V_5}$.

After finishing the subdivision process using feature values, the Doo-Sabin subdivision process will be carried out recursively on *surface faces* and on the edges of *plane faces*. The faces surrounding *plane faces* will be taken as open polygons. Figure 6 (c) illustrates the first Doo-Sabin subdivision process over the new polyhedron generated in Fig. 6 (b). Figure 6 (d) shows the second Doo-Sabin subdivision process. Figure 6 (e) shows the fourth Doo-Sabin subdivision process. It is clear that the two *plane faces* are getting closer to the faces of the original polyhedron with each step of the Doo-Sabin

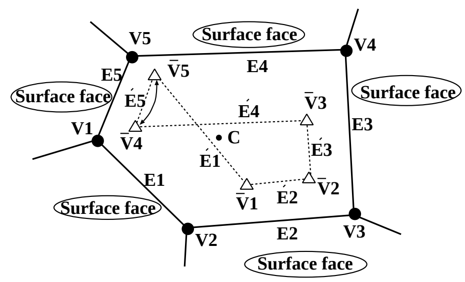


Fig. 7 Special case of the subdivision process using feature values.

subdivision process. The edges of the *plane face* are changed into biquadratic B-spline curves.

4. Examples

In this section, we take three examples. The first example, illustrated in **Fig. 8**, shows the effect of using different feature values in the subdivision process.

The second example models a surface over a club polyhedron. For comparison, **Fig. 9** illustrates the procedure of the Doo-Sabin subdivision process over a club-like polyhedron and the final club shapes, and **Fig. 10** illustrates the subdivision process with feature values over the same club polyhedron and the final club shapes. Observing the club with a roundish surface in Fig. 9 and the club in Fig. 10, the effect of subdividing the polyhedral network with feature values can clearly be seen. **Figure 11** illustrates the same process as Fig. 10. It shows that changing the FFV for just one face of the club polyhedron, from 1 to 0, yields a club with a plane face.

The third example, illustrated in **Fig. 12**, simulates a tube under different distortion pressures. Figure 12 (a) is the original polyhedron. Figure 12 (e) is the Doo-Sabin surface after four subdivision steps without assigning feature values. The other schemes are generated from the initial polyhedron by setting and adjusting feature values on corresponding vertices and edges. Obviously, the schemes generated with feature values are closer to reality.

The above three examples were generated for experiment. A simple interface was prepared to allow designers to select, input, and edit feature values of a polyhedron directly. The FFV,

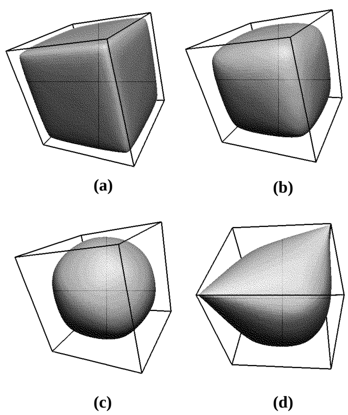


Fig. 8 A cube and surfaces generated using different feature values. All $FVFs$ of the faces of the cube are equal to 1 (*surface faces*). (a) All $EFVs$ on the faces of the cube are equal to 0.2, all $VFVs$ on the faces of the cube are equal to 1.0. (b) Changing all $EFVs$ on the faces of the cube to 0.4. (c) Changing all $EFVs$ on the faces of the cube to 0.6. (d) Changing $VFVs$ for two vertices on the faces that they are common to from 1.0 to 0.05.

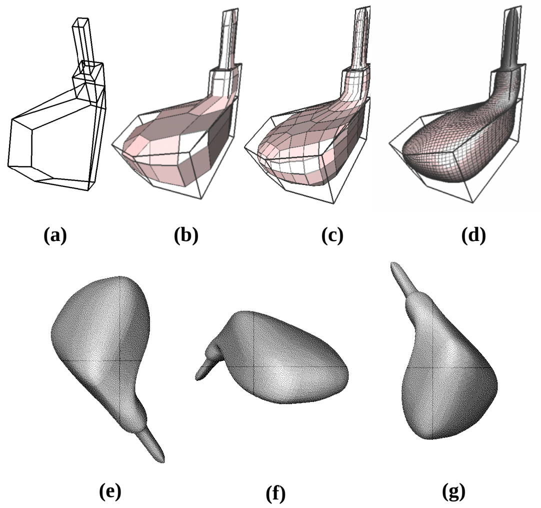


Fig. 9 A club polyhedron and the subdivision process using only the Doo-Sabin method. (a) A club polyhedron. (b) The first Doo-Sabin subdivision over the club polyhedron. (c) The second Doo-Sabin subdivision. (d) The fourth Doo-Sabin subdivision. (e)–(g) The final shapes from different viewpoints.

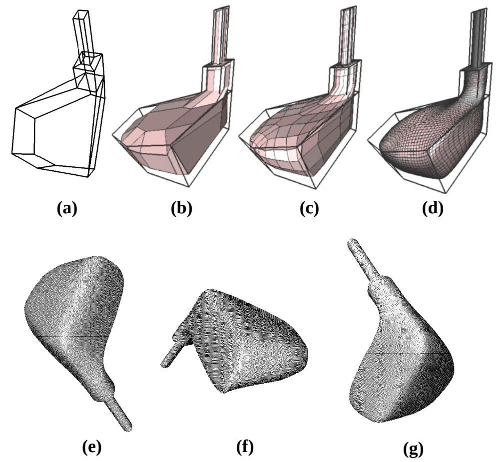


Fig. 10 A club polyhedron and subdivision process with feature values. (a) A club polyhedron. (b) The subdivision process with feature values over the club polyhedron. (c) The first Doo-Sabin subdivision over the polyhedron generated in step (b). (d) The third Doo-Sabin subdivision process. (e)–(g) The final shapes from different viewpoints.

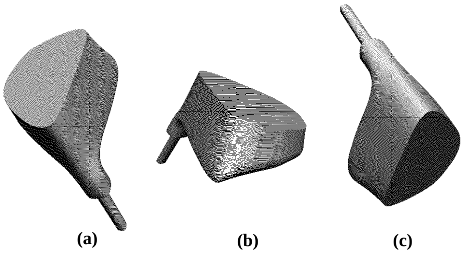


Fig. 11 The final club shapes with a *plane face*.

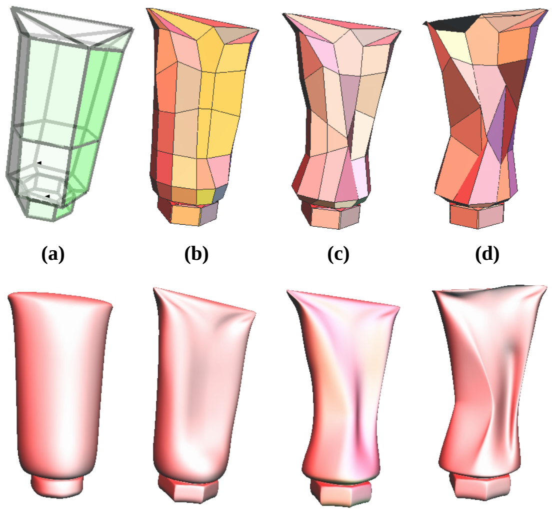


Fig. 12 Simulating a tube under different pressures. (a) The original polyhedron. (b)–(d) are the new polyhedra generated from the initial polyhedron in (a) with different feature values, respectively. (e)–(h) are the final shapes of (a)–(d), respectively.

EFV and VFV are defined in corresponding model data structure tables. Though the interface is very simple compared to creating these polyhedra directly using normal edit and draw functions, our method is still effective, as the generation process is done automatically.

5. Conclusions and Future Work

In this paper, we introduced the concept of feature values of a polyhedron for generating polyhedral networks effectively. Generally, designers can generate complex polyhedra using normal drawing and edit functions, but if feature values are introduced, designers can use parameters to control and model complex polyhedra from simple ones. It is clear that the input process can be more effective and intuitive. Moreover, though the process here is combined with the Doo-Sabin subdivision method, it also can be extended to other subdivision methods. The concept of feature values is also useful and helpful in CAGD systems, as creating and modifying polyhedra efficiently are basic requirements in these systems.

Of course, to introduce the technology proposed here into a system, there are many problems to be solved. For example, we need to study how to build the relationship of feature values and input method, how to create the data structure of a polyhedron and the interference problems of polyhedral networks or subdivision surfaces. Some basic problems of subdivision methods were studied in Ref. 4). Finally, we give our future work.

(1) The most important question is how to create an interactive interface that can allow input and editing of feature values of a polyhedron directly. For example, if we want to generate some graphs to simulate a tube under different pressures. The interface will be more friendly if we can build a suitable relationship between feature values and simulated input pressure. Then the final shape can be genetared easily by changing the pressures (Fig. 12).

(2) We need to be able to add a material property parameter to the feature value of a polyhedron, and set up a relationship between material property feature values. We can then generate complex shapes according to both the geometric and material properties of a model.

References

1) Chaikin, G.M.: An Algorithm for High Speed Curve Generation, *Comput. Gr. Image Process.*, No.3, pp.346–349 (1974).

2) Catmull, E.E. and Clark, J.H.: Recursively generated B-spline surfaces on arbitrary topological meshes, Comput. Aided Des., Vol.10, No.6, pp.350–355 (1978).

3) Doo, D. and Sabin, M.: Behaviour of recursive division surfaces near extraordinary points, *Comput. Aided Des.*, Vol.10, No.6, pp.356–360 (1978).

4) Nasri, A.H.: Polyhedral Subdivision Methods for Free-Form Surfaces, *ACM Trans. Graphics*, Vol.6, No.1, pp.29–73 (1987).

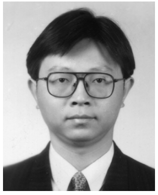
5) Nasri, A.H.: Surface interpolation on irregular networks with normal conditions, *Comput. Aided Geometric Des.*, Vol.8, pp.89–96 (1991).

6) Nasri, A.H.: Boundary-corner control in recursive subdivision surfaces, *Comput. Aided Des.*, Vol.23, No.6, pp.405–410 (1991).

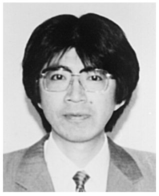
7) Nasri, A.H.: Curve interpolation in recursively generated B-spline surfaces over arbi-

trary topology, *Comput. Aided Geometric Des.*, Vol.14, No.1, pp.13–30 (1997).
8) Reif, U.: A unified approach to subdivision algorithms near extraordinary vertices, *Comput. Aided Geometric Des.*, Vol.12, pp.153–174 (1995).
9) Brunet, P.: Including shape handles in recursive subdivision surfaces, *Comput. Aided Geometric Des.*, Vol.5, pp.41–50 (1988).
10) Boehm, W., Farin, G. and Kahmann, J.: A survey of curve and surface methods in CAGD, *Comput. Aided Geometric Des.*, Vol.1, pp.1–60 (1984).
11) Tan, S.T. and Chan, K.C.: Generation of high order surfaces over arbitrary polyhedral meshes, *Comput. Aided Des.*, Vol.18, No.8, pp.411–423 (1986).
12) Dyn, N., Levin, D. and Gregory, J.A.: A Butterfly Subdivision Scheme for Surface Interpolation with Tension Control, *ACM Trans. Graphics*, Vol.9, No.2, pp.160–169 (1990).
13) Peters, J.: Smoothing Polyhedra Made Easy, *ACM Trans. Graphics*, Vol.14. No.2, pp.162–170 (1995).
14) Halstead, M., Kass, M. and DeRose, T.: Efficient, Fair Interpolation Using Catmull-Clark Surfaces, *Computer Graphics Proceedings*, *Annual Conference Series* (1993).
15) Zorin, D., Schroder, P. and Sweldens, W.: Interpolating Subdivision for Meshes with Arbitrary Topology, *Computer Graphics Proceedings*, *Annual Conference Series*, pp.189–192 (1996).
16) Farin, G.: *Curves and Surfaces for Computer Aided Geometric Design: A Practical Guide*, 4th Edition, Academic Press (1996).

(Received June 17, 1998)
(Accepted December 2, 1999)



Zheng Xu graduated from Shanghai University of Technology, China in 1985. He got his Master degree and Ph.D. degree from Saitama University, Japan in 1996 and 1999. His research includes computer graphics, the generation of free form surfaces and subdivision surfaces.



Kunio Kondo was born in 1954. He received his Ph.D. degree from The Univrsity of Tokyo in 1988. He had worked in Nagoya University since 1973. Since 1989, he has been in Saitama Univ. as an associate professor. His current research interests are Computer Graphics, Human Interface and Kansei Information Processing. He received the paper award of IPSJ in 1988. He is a member of IPSJ, ISGG, and JSGS.