2011 International Conference on Computer Science and Information Technology (ICCSIT 2011)
IPCSIT vol. 51 (2012) © (2012) IACSIT Press, Singapore
DOI: 10.7763/IPCSIT.2012.V51.79

Implementation of Triangle Subdivision for Holding Sharp Features with Flatness Control

Wei Liu[+], Lizhu Wang and Lina Xu

Faculty of Materials and Energy Guangdong University of technology Guangzhou(510006), P. R. China

Abstract. In this paper we introduce a improved rule for the well-known Loop subdivision that can keep the sharp shape of the object, in addition, we put forward a approach named flatness control to control how quickly the control points in an neighborhood converge towards the tangent plane which will obtain a more complicated surface. In order to get the required model, in the first step, we compute the position using the improved rules and modify it with flatness control in the second step.

Keywords: surface subdivision, Loop subdivision, sharp features, flatness control

1. Introduction

Since more than 20 years ago, subdivision surface technology has made great development mainly because it can generate smooth surface and be easy to realize for modal with arbitrary topological structure. Now subdivision surface modeling method has become one of the indispensable graphics modeling technology. A number of commercial systems use subdivision as surface representation: Alias|Wavefront's Maya, Pixar's Renderman, Autodesk's 3ds max, and Micropace' Lightwave 3D, to name just a few.

Although producing the smooth surface is a essential target for subdivision surface technology, some non-smooth effects are required in engineering surface modeling, such as crease, dart vertex and corner vertex, so how to keep these effects has become a topic recently. In existing feature modeling method, Hoppe[2] introduce a new method of piecewise smooth surface models based on Loop[3] subdivision, novel aspects of the method are its ability to model surfaces of arbitrary topological type and to recover sharp features such as creases and corners. However, the methods of Hoppe have two drawbacks: first, the corner rules of subdivision schemes can only generate convex corners. If the control mesh is in a concave configuration, the rules force the surface to approach the corner from the outer side, causing the surface to develop a fold; Second, The shape of the boundary of the generated surface depends on the control points in the interior which will develop gap between the meshes after subdivision. Biermann[4] present improved rules for Catmull-Clark and Loop subdivision that overcomes these above problems. Sederberg[5] develop rules for Non-uniform recursive Doo-Sabin and Catmull-Clark subdivision surfaces, a sharp parameter was introduced to control sharp effects, but it increase the complexity of the algorithm. Found that the degree of the parameter curve can decrease when insert repeated node, Wei[6] adjust the construction of mesh topology near the sharp feature to obtain sharp effects. Li [7] developed a fitting system to fit triangular meshes using Loop subdivision surfaces with sharp features; Liu [8] generate sharp features using - subdivision with sharp features; Zhou[9] propose a method for creating sharp features and semi-sharp features on hybrid subdivision surfaces which is introduced by Stam[10].

Because triangle meshes have a great power to represent many model, so in this paper, we introduce a improved rule for the well-known Loop subdivision that can keep the sharp shape of the object, in addition,

[+] Wei Liu. Tel.: 086-020-39322570; fax: 086-020-39322570
E-mail address: collionliu@163.com

we put forward a approach named flatness control to control how quickly the control points in a neighborhood converge towards the tangent plane which will obtain a more complicated surface.

2. The Loop subdivision for holding sharp features

2.1. Sharp Feature

Hoppe[2] at el. defined that a crease is a tangent line smooth curve along which the surface is C0 but not C1 and also classify sharp feature vertices into three different types:

(1) A corner vertex is a point where three or more creases meet;

(2) A dart vertex is an interior point of a surface where a crease terminates;

(3) A crease vertex joints exactly two incident crease edges smoothly;

Biermann[4] at el. specify all edges on the boundary of the mesh are crease edges, boundary vertices are corner or crease vertices. Creases divide the mesh into separate patches, several of which can meet in a corner vertex. At a corner vertex, the creases separate the ring of triangles into several sectors. They label each sector of the mesh as convex sector (where the angle between two crease edges is less than π) or concave sector (where the angle between two crease edges is more than π). An example of a tagged mesh is given in Fig. 1.

2.2. The extended loop's scheme

Loop's scheme is a generalization of C^2-continue quartic triangular B-spline. The refinement step proceeds by splitting each triangular face into four subfaces. The vertices of the refined mesh are then positioned using weighted averages of the vertices in the unrefined mesh. Formally, starting with the initial control mesh $M = M^0$, each subdivision step carries a mesh $M^r = (K^r, V^r)$ into a refined mesh $M^{r+1} = (K^{r+1}, V^{r+1})$ where the vertices V^{r+1} are computed as affine combinations of the vertices of V^r. Some of the vertices of v^{r+1} naturally correspond to vertices of V^r —these are called vertex points; the remaining vertices in v^{r+1} correspond to edges of the mesh M^r — these are called edge points. Let v^r denote a vertex of V^r having neighbor $v_1^r,\cdots,v_n^r$ as shown in Fig. 2, such a vertex has valence n Let v_i^{r+1} denote the edge point of Vr+1 corresponding to the edge vrvir, and let vr+1 be the vertex point of Vr+1 associated with vr. The positions of vr+1 and vir+1 are computed according to the subdivision rules. As the speed of the control points in a neighborhood converging towards the tangent plane is related to the limited position and tangent vector of Loop subdivision, and these results can be achieved according to control points in a the neighborhood, here we also offer these computed rules.

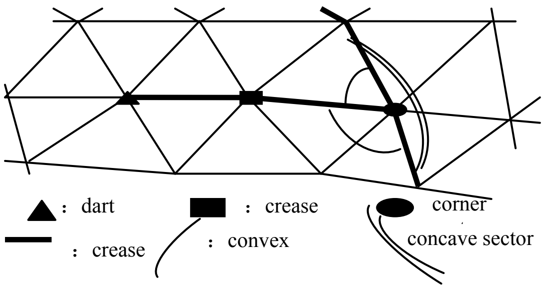


Fig1. a tagged mesh

2.2.1 The rule of smooth or dart vertices

For smooth or dart vertices, computed rules of associative values are written as follows (the mask see Fig. 3), n is the number of edges adjacent to this vertex.

(1) New smooth or dart vertex point:

$$v^{r+1} = (1-n\beta)v^r + \beta\sum_{i=1}^{n} v_i^r \tag{1}$$

Ensure that you return to the 'Els-body-text' style, the style that you will mainly be using for large blocks of text, when you have completed your bulleted list.

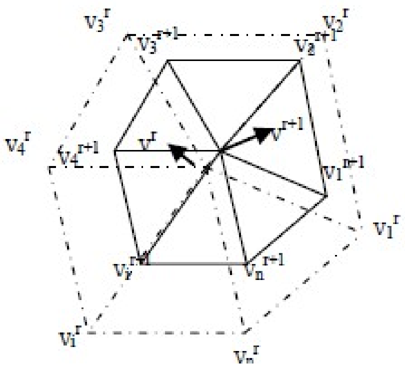


Fig.2. The neighborhood around a vertex of valence n

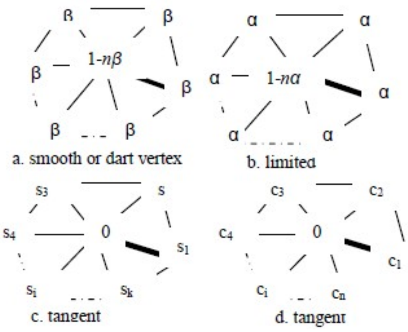


Fig.3. smooth or dart

(2) Limited position:

$$v^{\infty} = (1 - n\alpha)v' + \alpha\sum_{i=1}^{n} v_i^r \tag{2}$$

Where for n=3, α = 1/5; For n≠3, α= 1/(2n)

Tangent vector in one direction:

$$l_1 = \sum_{i=1}^{n} s_i v_i^r\ ) \tag{3}$$

Where $s_i = 2\sin(2\pi i/n)/n$

(3) Tangent vector in another direction:

$$l_2 = \sum_{i=1}^{n} c_i v_i^r \tag{4}$$

Where $c_i = 2\cos(2\pi i/n)/n$

2.2.2 The rule of crease vertices

For crease vertices and boundary vertices, we apply same rules. For a crease vertex, the two incident crease edges divide its neighborhood into two different parts, here we adopt the follow rules to compute our needed value of each part respectively (the mask see Fig. 4), where vrv0r and vrvmr are crease edges, m is the numbers of triangles of one part.

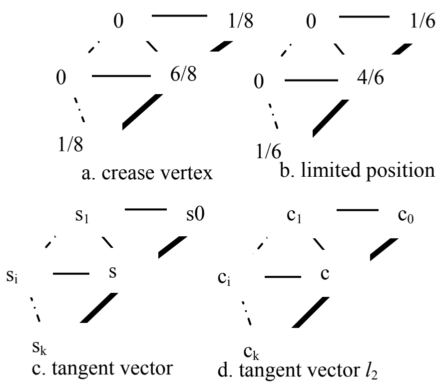


Fig.4. crease vertex mask

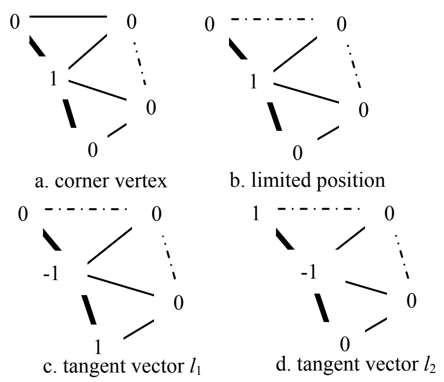


Fig.5. corner vertex mark

(1) New crease vertex point:

$$v^{r+1} = (6v^r + v_0^r + v_m^r)/8 \tag{5}$$

(2) Limited position:

$$v^{\infty} = (4v^r + v_0^r + v_m^r)/6\ ) \tag{6}$$

(3) Tangent vector in one direction:

$$l_1 = sv' + s_0 v_0^r + s_m v_m^r + \sum_{i=1}^{m-1} s_i v_i^r) \quad . \quad . \quad . \quad . \quad . \tag{7}$$

Where for m=1, s=-1, $s_0 = s_1$ = 1/2;

For $m \neq 1$, s=0, s_0=1/2, s_m = -1/2, s_i = 0

(4) Tangent vector in another direction:

$$l_2 = cv' + c_0 v_0^r + c_m v_m^r + \sum_{i=1}^{m-1} c_i v_i^r) \quad . \quad . \quad . \quad . \quad . \quad . \quad . \quad . \quad . \tag{8}$$

Where for m = 1, c = 0, c_0 = 1/2, c_1 = -1/2;

For $m \neq 1$,

$$c = -2 \cdot [(2/3 - a)\sigma_1 - b\sigma_3]/m \ ,$$

$$c_0 = c_m = -2 \cdot [(a/2 + 1/6)\sigma_1 + b\sigma_2/2]/m \ ,$$

$$c_i = 2\sin(\pi i/m)/m \ , \quad i = 1, \ldots, m-1$$

$$a = [0.25(1 + \cos(\pi/m)]/[1.5 - 0.75\cos(\pi/m)] \ , \quad b = (2/3 - a)/\cos(m\zeta/2)$$

$$\sigma_1 = \sin(\pi/m)/[1 - \cos(\pi/m)] \ ,$$

$$\sigma_1 = [\cos(m\zeta/2)\sin(\pi/m)]/[\cos\zeta - \cos(\pi/m)] \ ,$$

$$\zeta = \arccos(\cos\pi/m - 1)$$

2.2.3 The rule of corner vertices

For a corner vertex, the several crease edges divide its neighborhood into several different parts, as each part have different control points, the computed result of each part are different accordingly. The computed formulas of the certain part near a corner vertex are written as follows (see Fig. 5), here vrv0r and vrvtr are crease edges, and t is the numbers of triangles adjacent to the corner vertex between two crease edges.

(1) New corner *vertex point:* $v^{r+1} = v^r$. (9)

(2) Limited *position:*

$$v^{\infty} = v^r \quad . \quad . \quad . \quad . \quad . \quad . \tag{10}$$

(3) Tangent vector in one direction:

$$. \quad l_1 = -v' + v_0^r \quad . \quad . \quad . \quad . \quad . \quad . \tag{11}$$

(4) Tangent vector in another direction:

$$l_2 = -v' + v_i^r \tag{12}$$

2.2.4 The rule of smooth edge

The computed rule of new edge points is the most complicated (see Fig. 6 and Fig. 7), because it depend on the style of the edges, the style of the two endpoints and sectors. If the edge isn't a crease edge, according to the tags of the two endpoints, we list two rules to computer new smooth edge points (see Fig. 6a and Fig. 6b). After refinement, the old edge will be split into two new smooth edges (not tagged), and the new edge point will be labeled as a smooth point.

(1) New smooth edge point:

①In the absence of any special tags, we apply the standard edge rules. The averaging masks are given in Fig. 6a.

$$v_i^{r+1} = (3v^r + 3v_i^r + v_{i-1}^r + v_{i+1}^r)/8 \tag{13}$$

②In the case of an untagged edge $v^r v_i^r$ adjacent to a tagged vertex v^r is illustrated in Fig. 6b.

$$v_i^{r+1} = [(6 - 8\lambda)v^r + 8\lambda v_i^r + v_{i-1}^r + v_{i+1}^r]/8 \tag{14}$$

If v^r is dart vertex, λ = 1/2—cos(2π/k)/4; If v^r is a crease vertex, λ = 1/2—cos(π/k)/4; For a convex corner λ = 1/2—cos(φ/k)/4, if e is in a convex part φ = π/2,otherwise φ = 3π/2.

(2) Limited position:

$$v_i^{\infty} = v_{i+1}^r + 5(v^r + v_i^r + v_{i-1}^{r+1} + v_{i+1}^{r+1} + v_{e1}^{r+1} + v_{e2}^{r+1})/6 \tag{15}$$

(3) Tangent vector in one direction:

$$l_1 = (v_{i-1}^{r+1} - v_{i+1}^{r+1} + v_{e1}^{r+1} - v_{e2}^{r+1}) \cdot \sin(\pi/3) \tag{16}$$

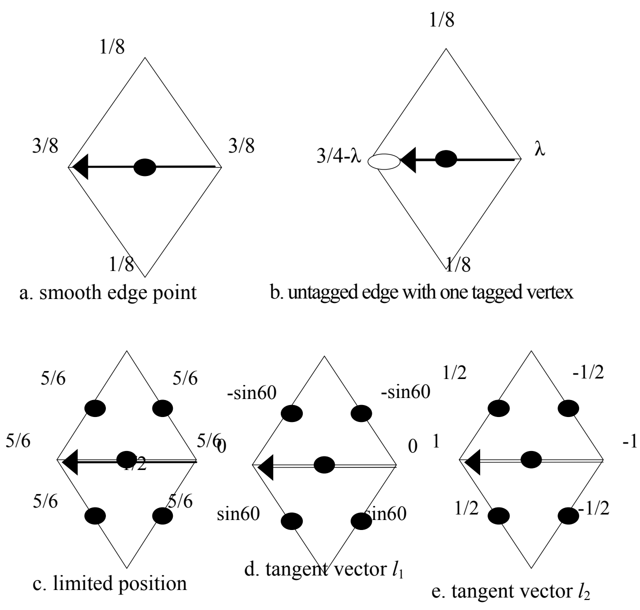


Fig.6. smooth edge mark

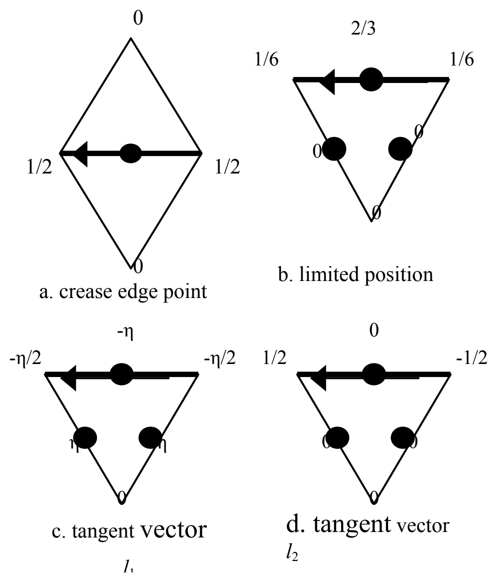


Fig.7. crease edge mark

(4) Tangent vector in another direction:

$$l_2 = v^r - v_i^r + (v_{i-1}^{r+1} + v_{i+1}^{r+1} - v_{e1}^{r+1} - v_{e2}^{r+1})/2 \tag{17}$$

2.2.5 The rule of crease edge

For a crease edge, we insert a new vertex on it as the average of the two adjacent vertices (given in Fig. 7a), then the two new edges are tagged as crease edges, the new vertex is tagged as crease vertex. The rules of all needed value are shown as follow (the mask see Fig. 7):

(1) New crease edge point:

$$v_i^{r+1} = v^r/2 + v_i^r/2 \tag{18}$$

(2) Limited position:

$$v_i^\infty = v^r/6 + v_i^r/6 \qquad (19)$$

(3) Tangent vector in one direction:

$$l_1 = -\eta v_i^{r+1} + \eta(-v^r/2 - v_i^r/2 + \eta v_{i-1}^{r+1} + \eta v_{e1}^{r+1}) \qquad (20)$$

Where $\eta = 0.288675$.

(4) Tangent vector in another direction:

$$l_2 = v^r/2 - v_i^r/2 \qquad (21)$$

In the (13) ~ (21) for Fig. 6 and Fig. 7, the vertex at which the arrow point represent v^r, the neighborhood of v^r see in Fig. 2, let n is the valence of v^r, $i = 1, \ldots, n$, when subscript is less than 1, we use n instead of it, when subscript is more than n, we use 1 instead of it, v_i^{r+1} is the edge point corresponding to the edge $v^r v_i^r$, v_{e1}^{r+1} is the edge point corresponding to the edge $v_{i-1}^r v_i^r$, v_{e2}^{r+1} is the edge point corresponding to the edge $v_i^r v_{i+1}^r$.

3. Flatness modification

It is observed that the speed that the control points in a neighborhood converge towards the tangent plane can be control. The control equation is

$$p^{new} = (1-f)p + f(u_0 + x\sigma_1 u_1 + y\sigma_2 u_2) \qquad (22)$$

$$p_i^{new} = (1-f)p_i + f(u_0 + x\sigma_1 u_1 + y\sigma_2 u_2) \qquad (23)$$

Equation (22) is applied to modify position of vertex points of one refined mesh, (26) is used to modify position of edge points of one refined mesh.

In the (22) and (23), f is a flatness modified parameter, and $0 \le f \le 1$, the default values for the parameter f is 0.5; u_0 is the limited position of the control point; u_1 and u_2 are the tangent vectors in the two direction; In the (22), p is position vector of control point after refinement, if the control point is a dart vertex of unrefined mesh, $\sigma_1 = \sigma_2 = 3/8+\cos(2\pi/n)/4$ (where n is the number of triangles adhere to this control point), $x=y=0$; if it is a crease vertex of unrefined mesh, $\sigma_2 = 0.5$, if $n=1$, $\sigma_1 = 0.25$, $x=1/3$, $y=0$; if $n \neq 1$, $\sigma_1 = 0.5$, $x=y=0$;if v correspond to a corner vertex of unrefined mesh $\sigma_1 = \sigma_2 = 0.5$, $x=y=0$; In the (23), $\sigma_1 = \sigma_2 = 0.5$, pi is position vector of edge point which associated with edge $v^r v_i^r$ in the neighborhood of v^r, when v^r is dart vertex, $x=\sin 2\pi i/n$, $y= \cos 2\pi i/n$, when v^r is a crease vertex, $x=\sin \pi i/n$, $y= \cos \pi i/n$, when v^r is corner vertex, $x_i = \sin[(k-i)\varphi/k]/\sin\varphi$, $y_i = \sin(i\varphi/k)/\sin\varphi$, For a convex corners we use $\varphi = \pi/2$,For concave corners $\varphi = 3\pi/2$.

The modified rule blends between control point positions before flatness modification and certain points in the tangent plane, which are close to the projection of the original control point. The flatness modification is always applied at concave corner vertices and boundary vertices to make sure the surface is smooth around the control points.

In the first step, we compute the position using the improved rules as above and modify it with flatness control in the second step.

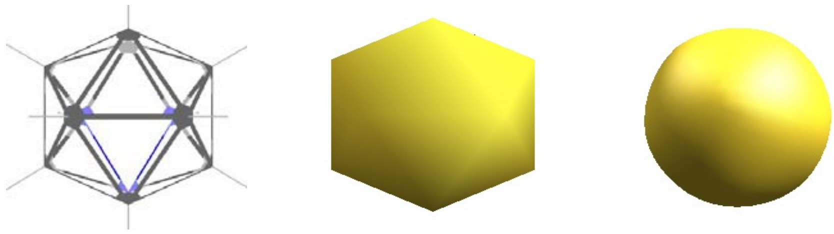
4. Results and conclusion

Surfaces with creases and corners of various types are illustrated in Fig. 7 to Fig. 11. All the surfaces in these figures are generated from the same control mesh by applying different tags. In Fig. 8, the object carry a corner vertex, a crease vertex and crease edges, In Fig. 9, the object carry a dart vertex, a crease vertex and crease edges, In Fig. 10, the object carry boundary edges. Figures 11 demonstrate flatness modification for boundary and interior vertices .Note the different result of this modal.

Conclusions: Here are some findings from these figures:

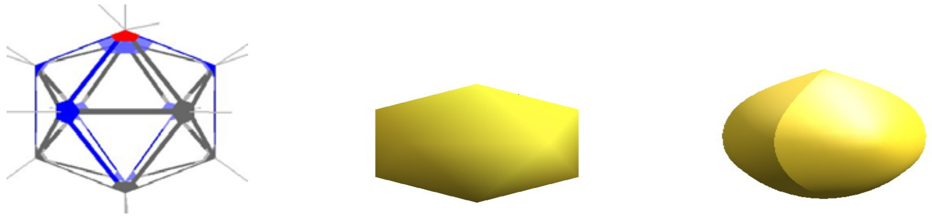
(1) Along the crease, the surface is C0 continuous, and the corner vertex is interpolated.
(2) At the dart vertex, crease link to surface smoothly.
(3) With the increase of parameter f, the special tagged sector will approach toward the tangent plane of the control point more quickly.

So we have given a simple modification of the most popular subdivision scheme that can keep sharp feature effectively and provided flatness control which can create more characteristics for surface modeling.



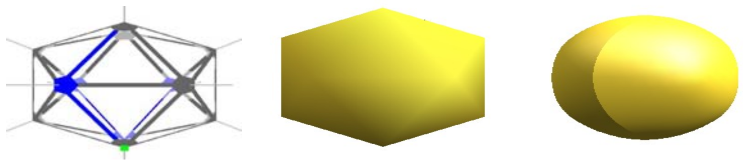
(a) Initial control mesh (b) Initial result (c) subdivision result

Fig 8. regular subdivision



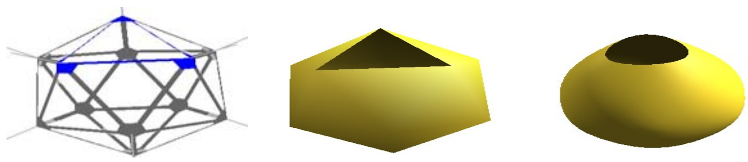
(a) Initial control mesh (b) Initial result (c) subdivision result

Fig 9. crease and corner vertices



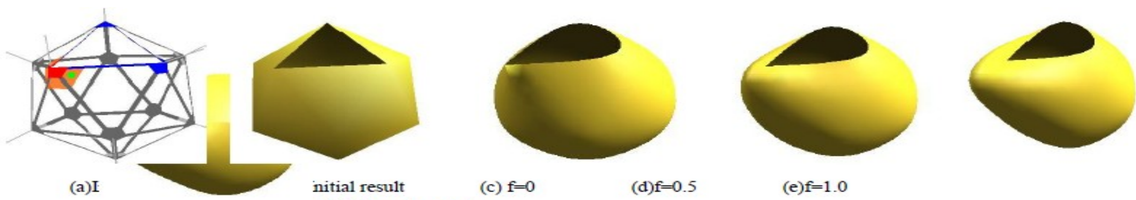
(a) Initial control mesh (b) Initial result (c) subdivision result

Fig 10. crease and dart vertices



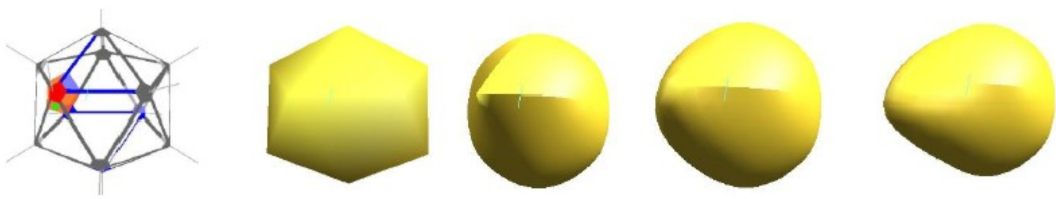
(a) Initial control mesh (b) Initial result (c) subdivision result

Fig. 11 boundary edges



(a)I nitial result (c) f=0 (d)f=0.5 (e)f=1.0

（1） boundary vertex



(a) Initial control mesh (b) Initial result (c) f=0 (d) f=0.5 (e) f=1.0

（2） interior vertex

Fig. 12： flatness modification

5. Acknowledgements

The Project-supported by Natural Science Foundation of Guangdong Province, China (8451009001001237) and the crucial field and key breakthrough project in Guangdong-Hongkong (2009Z019)

6. References

[1] DeRose T, Kass M, Tien T. "Subdivision surface in character animation," Computer Graphics Proceeding, Annual Conference Series, ACM SIGGRAPH. New York: ACM Press, pp.85~94, 2000 .

[2] Hoppe H, DeRose T, Duchamp T, et al. "Piecewise smooth surface construction". Computer Graphics Proceedings, Annual Conference Series, ACM SIGGRAPH. Orlando, Florida: ACM Press, pp.295~ 302, 1994.

[3] Loop C. Smooth subdivision surfaces based on triangles. Utah, USA: University of Utah, Department of Mathematics, 1987.

[4] Biermann H, Levin A, Zorin D. "Piecewise Smooth Subdivision Surfaces with Normal Control." Computer Graphics Proceedings, Annual Conference Series, New York; ACM, pp.113~120, 2000.

[5] Sederberg T, Zheng J, Swell D, et a1. "Non-uniform recursive subdivision surfaces". Computer Graphics Proceedings, Annual Conference Series, ACM SIGGRAPH, Orlando, Florida, pp395 – 404, 1998.

[6] Wei Guofu, Chen Falai. "Sharp features of subdivision surfaces over triangular meshes," Journal of China University of Science and Technology, Vol. 32, pp 140~146, 2002.

[7] Li Guiqing, Ma Weiyin, Bao Hujun. "Fitting System Using Loop Subdivision Surfaces with Sharp Feature," Journal of Computer Aided Design and Computer Graphics, Vol. 17, pp. 1180-1185, 2005.

[8] Liu Ying, Liu Ze-yi, Gao Pengdong. "A -Subdivision Algorithm for Holding Sharp Features," Chinese Journal of Computer, Vol 28, pp.1398-1403, 2005.

[9] Zhou Min, Peng guohua, Ye Zhenglin at el. "Hybrid Subdivision Algorithm with Special Effect," Journal of Computer Aided Design and Computer Graphics, Vol. 19, pp. 786-791, 2007.

[10] Stam J, Loop C. "Quad/triangle subdivision. Computer Graphics Forum," 2003, Vol. 22, 79-85, 2003.