www.cadanda.com Computer-Aided Design AND APPLICATIONS

A Heuristic Offsetting Scheme for Catmull-Clark Subdivision Surfaces

Jianzhong Wang[1] and Fuhua Cheng[1,2]

[1]University of Kentucky, jwangf@uky.edu, cheng@cs.uky.edu
[2]National Tsing Hua University, cheng@cs.uky.edu

ABSTRACT

In rapid prototyping, a hollowed prototype is preferred and significantly reduces the building time and material consumption in contrast to a solid model. Most rapid prototyping obtains solid thin shell by gradually adding or solidifying materials layer by layer. This is a non-trivial problem to offset a solid which involves finding all self-intersections and filling gaps after raw offsetting. While Catmull-Clark subdivision (CCS) surfaces are widely used in solid modeling, the hollow solid/thin shell problems are not well addressed yet. In this paper, we explore earlier methods of obtaining thin shell CCS solid and present a new thin solid approach. With this new scheme, one can efficiently avoid creases and handle gaps.

The new scheme is heuristic, but inner surface is parametric, so computation of the inner surface is simplified. And with offsetting Bezier crust applied, the inner surface maintains the mesh structure and continuity of the outer surface. The obtained thin shell solid is C^2 continuous everywhere, except at extraordinary points, where it is C^1 continuous.

Keywords: rapid prototyping, hollow out a solid, 3D surface offsetting, Bezier crust, Catmull-Clark subdivision.

1. INTRODUCTION

Catmull-Clark subdivision (CCS) surfaces [1] have been widely used in computer graphics and animation. In contrast to traditional spline schemes, the CCS scheme can handle arbitrary topology and is easy to design and implement. In 3D modeling, building a hollowed prototype instead of a solid model is required to reduce the building time and material consumption. When we use CCS to generate a hollowed object, the intuitive way is to construct CCS meshes for both the outer and the inner surfaces. However, it is not effective and many issues arise during construction of the inner surface, e.g., surface collision, self-intersections. It is not an easy task to design a CCS control mesh to generate a thin-shell hollowed 3D object.

In CAD/CAM, rapid prototyping (RP) builds a part layer by layer faster than traditional prototyping methods. The RP process involves slicing the CAD model perpendicular to the building direction sequentially and gradually adding or solidifying materials layer by layer. RP applications are used in the making of molds, manufacturing parts, and most recently 3D-printing. In RP, when each layer is solid, it not only consumes more materials, but also is time consuming. To reduce the building time and material consumption, the method of hollowing out the 3D solids is applied to reduce the cross-sectional area to be traced.

Some spatial enumerations have been used to obtain hollow solids, such as a sub-boundary octree [10] located inside the original solid, voxel model [2] featuring one-dimensional Boolean operations between the ray representation and voxel elements. The main problem with enumeration techniques is the staircase effect, which make offsetting surface not attractive.

Another method developed is constructive solid geometry (CSG) [11]. CSG works by subtracting the original solid from its offset counterpart. This method is known to perform well on simple primitives, such as cylinder, spheres and boxes. However it is difficult to offset a free-form surface like CCSS.

2D curve offsetting method [5] slices the original solid sequentially and obtains internal cross-sectional curves by offsetting external cross-sectional curves of each slice. This method is simple and easy to implement, but it is hard to achieve uniform wall

Taylor & Francis
Taylor & Francis Group

thickness. A further work [13] achieves more uniform wall thickness and proposed a new algorithm that computes internal contour without computing the offset model.

There are also some surface offsetting methods. Non-uniform offsetting method [8] employs a vertex offsetting approach which is based on an averaged surface normal method. Main issue with this method is the existing of many self-intersections and invalid triangles. Computing the correct offset model of a STL model is a non-trivial task [12].

Several isocurve-based methods are developed to offset free-form surfaces. These methods are based on 3D curve offsetting [14]. In methods of tool-path generation [4] and adaptive isocurve-based rendering [3], a set of parallel curves called iso-distance curves are obtained by trimming iso-parametrics situated at fixed distances from the original curves. An iterative method of interference-free 3D offset contours [6] is proposed to offset parametric surfaces.

Given a free-form parametric surface like CCSS, if we apply above methods, although 3D offset surface generated will maintain uniform wall thickness, but the surface quality will not be satisfactory. None of above can generate an C^2 offset surface. It will be acceptable if there is no surface quality requirement for the offset surface. However, when the model is used to make mold, it is generally required the 3D offset surface is also smooth.

In this paper, we present an C^2 offsetting scheme on CCS surfaces. With this new scheme, one can generate hollow 3D solids efficiently with one layer of CCS control mesh and maintain the curvature continuity of CCS scheme. Due to the parametric properties of CCS, in our new scheme, we use a new surface offsetting approach, which offsets the limit surface directly by adding a thin layer of bi-quintic Bezier surface. Fig. 1 shows a hollowed solid after applying our new scheme, from 1(c) and (e) we see that the offsetting surface is smooth and the wall thickness is visually uniform.

The remainder of the paper is organized as follows. Section 2 discusses earlier works, section 3 presents our new offsetting scheme with Bezier crust, and section 4 shows behavior of our new scheme with illustrative examples. Finally, concluding remarks are made in section 5.

2. EARLIER WORKS

A Catmull-Clark subdivision (CCS) surface is the limit surface of a sequence of subdivision steps performed on a given control mesh. At each step, new vertices are added and old vertices are updated. The *valence* of a vertex is the number of edges meeting at the vertex. A vertex with valence four is called a regular vertex, otherwise an extraordinary vertex. A mesh face is regular if all vertices are regular, otherwise, it is called extraordinary face. CCS vertices are classified into three categories: *vertex points, edge points,* and *face points*. A popular way to index the control vertices is shown on the left side of Fig. 2 for a regular face and the right side for an extraordinary face, where V is a vertex point, E_i's are edge points, F_i's are face points, and $I_{i,j}$'s are inner ring control vertices. New vertices within each subdivision step are generated as follows:

$$
\begin{aligned}
V' &= \alpha_N V + \beta_N \sum_{i=1}^{N} \frac{E_i}{N} + \gamma_N \sum_{i=1}^{N} \frac{F_i}{N} \\
E_i' &= \frac{3}{8}(V + E_i) + \frac{1}{16}(E_{i+1} + E_{i-1} + F_i + F_{i-1}) \\
F_i' &= \frac{1}{4}(V + E_i + E_{i+1} + F_i)
\end{aligned}
\tag{2.1}
$$

Where N is the valence of vertex V, with $\alpha_N = 1 - \frac{7}{4N}$, $\beta_N = \frac{3}{2N}$, $\gamma_N = \frac{1}{4N}$.

The CCS limit surface obtained by performing (2.1) sequentially can be parameterized [9]. We define $S(u,v)$ as the CCS limit surface with parametric values (u,v), $u,v \in [0,1]$, such that the CCS limit/data point $S(0,0)$ of vertex point V is

$$
S(0,0) = \frac{5V + (12\beta_N + 8\gamma_N)E + (2\beta_N + 8\gamma_N)F}{5 + 14\beta_N + 16\gamma_N} \tag{2.2}
$$

Where $E = \left(\sum_{i=1}^{N} E_i\right)/N$, $F = \left(\sum_{i=1}^{N} F_i\right)/N$.

The unit normal $\vec{n}_{S(u,v)}$ on each data point $S(u,v)$ of CCS limit surface can be explicitly calculated with its first order partial derivatives $\frac{\partial S(u,v)}{\partial u}$ and $\frac{\partial S(u,v)}{\partial v}$,

$$
\vec{n}_{S(u,v)} = \frac{\dfrac{\partial S(u,v)}{\partial u} \times \dfrac{\partial S(u,v)}{\partial v}}{\left| \dfrac{\partial S(u,v)}{\partial u} \times \dfrac{\partial S(u,v)}{\partial v} \right|}. \tag{2.3}
$$

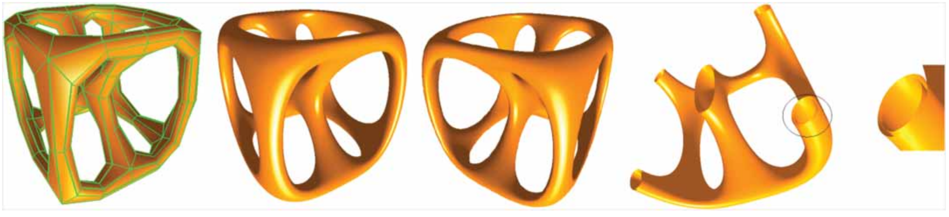

Fig. 1: an example of hollowed solid with our new offsetting scheme: a) CCS mesh, b) CCS limit surface, c) our offsetting surface, d) cross-section view, e) enlarged detail from cross-section.

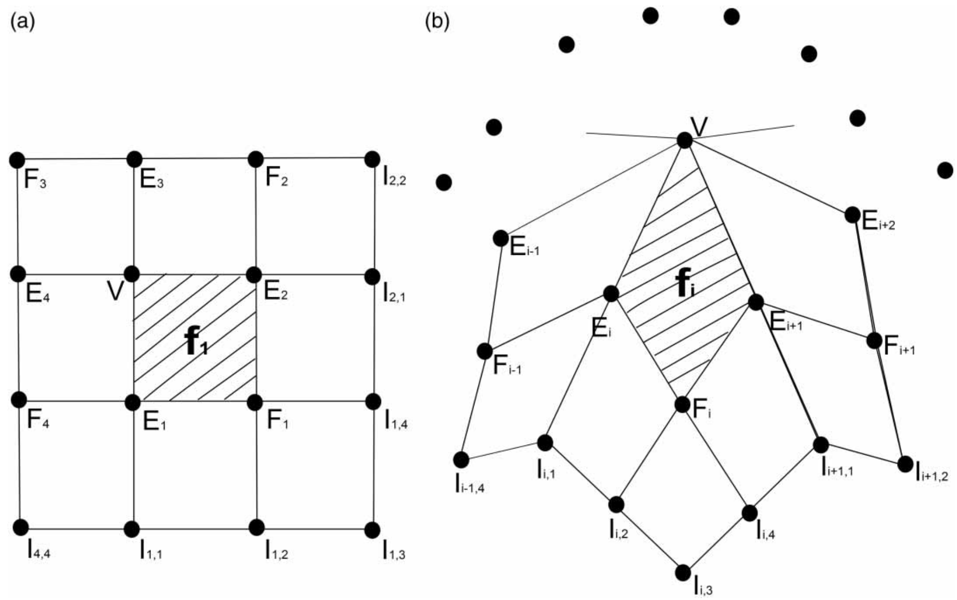

Fig. 2: mesh structure of CCS, (a) regular face, (b) extraordinary face.

Given an offset thickness d, the simplest solution of constructing an offset surface $\bar{S}(u,v)$ is to subtract from each data point $S(u,v)$ a vector of size d along the direction of unit normal,

$$\bar{S}(u,v) = S(u,v) - d \cdot \vec{n}_{S(u,v)} \tag{2.4}$$

This scheme works fine when the limit surface is concave or flat, but it will possibly generate a creased surface when the limit surface is convex. The creases (self intersections) (Fig. 3 (b)) are caused by intersection of $\vec{n}_{S(u,v)}$'s of neighboring data points along the surface, which can only be reduced by decreasing the number of data points on each face (Fig. 3 (c)) or shortening the offset thickness d (Fig. 3 (d)). However, since less number of data points means more roughness of limit surface and the reduction of thickness d is usually unwanted, the creases cannot be effectively removed.

In [15], a Bezier crust scheme is applied to CCS limit surface to obtain a parametric interpolating surface, the bi-quintic Bezier crusts added will maintain the curvature continuity of underlying CCS parametric surfaces. This is consistent with research of [7]. The scheme of Bezier crust works on difference vectors between control points and their corresponding data points.

In the Bezier crust scheme, given a quad control mesh M, the CCS scheme generates a limit surface that approximates the control mesh. The limit surface of each face f of M (regular or extraordinary) can be represented in parametric form $S(u,v)$. For each f, ΔP_0, ΔP_1, ΔP_2 *and* ΔP_3 are defined as the difference vectors between the corner control points and their corresponding CCS data points, respectively. In order to interpolate the control points, a bi-quintic Bezier crust $\Delta p(u,v)$ is defined as follows,

$$\Delta p(u,v) = \sum_{i=0}^{5}\sum_{j=0}^{5} \mathrm{b}_{\mathrm{i},5}(\mathrm{u})\mathrm{b}_{\mathrm{j},5}(\mathrm{v})\Delta P_{i,j} \tag{2.5}$$

With $\Delta P_{i,j}$ takes value of ΔP_0, ΔP_1, ΔP_2 *and* ΔP_3. $\Delta P_{i,j} = \Delta P_0$ if $\in [0,2]$ & $j \in [0,2]$; $\Delta P_{i,j} = \Delta P_1$ if $i \in$

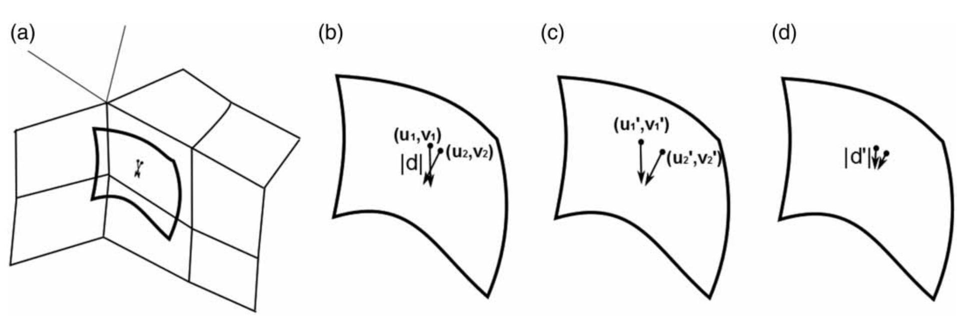

Fig. 3: (a) a CCS control mesh and limit face; (b) the two neighboring offsetting data points crease; (c) solve creasing issue by decreasing the number of data points; (d) solve creasing issue by a shorter $|d|$.

$[0,2]$ & $j \in [3,5]$; $\Delta P_{i,j} = \Delta P_2$ if $i \in [3,5]$ & $j \in [0,2]$; $\Delta P_{i,j} = \Delta P_3$ if $i \in [3,5]$ & $j \in [3,5]$. ΔP_0, ΔP_1, ΔP_2 *and* ΔP_3 are the difference vectors at four corners of a CCS face (Fig. 4) .

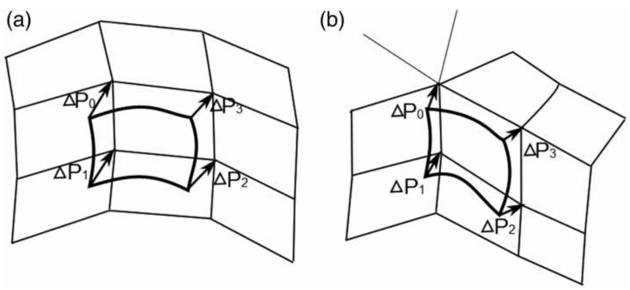

Fig. 4: difference vectors of ΔP_0, ΔP_1, ΔP_2 *and* ΔP_3 on (a) a regular face and (b) an extraordinary face.

An interpolating surface constructed by appending a bi-quintic Bezier crust on CCSS (shown in (2.5)) has the following properties:

- It interpolates exactly the corner control points
- It maintains the CCSS 1^{st} and 2^{nd} order derivatives at the corner control points
- It is C^2 continuous everywhere, except at extraordinary points, where it is C^1 continuous

Figure 5 shows that the interpolating surface is smooth and appropriate for most engineering/CAD usage. This inspired our interest to apply Bezier crust on CCSS to obtain a hollowed solid, such that a smooth offsetting surface can be constructed similar to CCSS, while maintains the curvature continuity of original CCS surfaces.

3. OFFSETTING SURFACE ON CCSS WITH BEZIER CRUST

In previous section, we introduced an interpolating scheme with Bezier crust on CCSS. In this section, we show how this scheme can be applied to construct a smooth offsetting surface on CCS surfaces.

Given a CCS control mesh M, on an arbitrary face, we define f with a set of $2N+8$ control points $V, E_1, \ldots, E_N, F_1, \ldots, F_N, I_1, \ldots, I_7, i = 1, ..N$ (shown in Fig. 2). With parametric form $S(u,v)$ of CCS, we define the data points at four corners of CCS limit surface as $p_0 = S(0,0), p_1 = S(1,0), p_2 = S(1,1), \wedge p_3 = S(0,1)$, and their unit normal as $\overrightarrow{n_i}$ (Fig. 6).

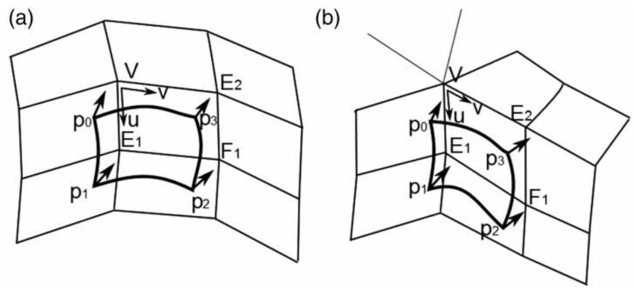

Fig. 6: CCS control mesh and its corner data points and normals: (a) regular face, (b) extraordinary face.

If we set the desired thin-shell thickness as d, then we can define a set of difference vectors of Δp_i on their corresponding data point p_i, then $(p_i - \Delta p_i)$ will be the desired corner data points on the offsetting surface. When we apply Bezier crust on these difference vectors at four corners of a CCS face, we can obtain a parametric offsetting surface having uniform distance of d at all corners of each CCS face with its corresponding CCS corner points, while CCS continuity will be kept after offsetting. Our scheme select Δp_i with

$$\Delta p_i = d \cdot \overrightarrow{n_i} \tag{3.1}$$

The computation of the four corner points on the new offsetting surface is consistent with the method used in method of 3D surface offsetting in [8].

With (3.1), we now define the offsetting Bezier crust $\Delta S(u,v)$ on difference vectors of Δp_i $(i = 1, \ldots, 4)$, with expression of

$$\Delta S(u,v) = \sum_{i=0}^{5} \sum_{j=0}^{5} \mathrm{b_{i,5}(u) b_{j,5}(v)} \Delta P_{i,j} \tag{3.2}$$

With $\Delta P_{i,j}$ takes value of Δp_0, Δp_1, Δp_2 *and* Δp_3. $\Delta P_{i,j} = \Delta p_0$ if $\in [0,2]$ & $j \in [0,2]$; $\Delta P_{i,j} = \Delta p_3$ if $i \in [0,2]$ & $j \in [3,5]$; $\Delta P_{i,j} = \Delta p_1$ if $i \in [3,5]$ & $j \in [0,2]$; $\Delta P_{i,j} = \Delta p_2$; if $i \in [3,5]$ & $j \in [3,5]$. Δp_0, Δp_1, Δp_2 *and*

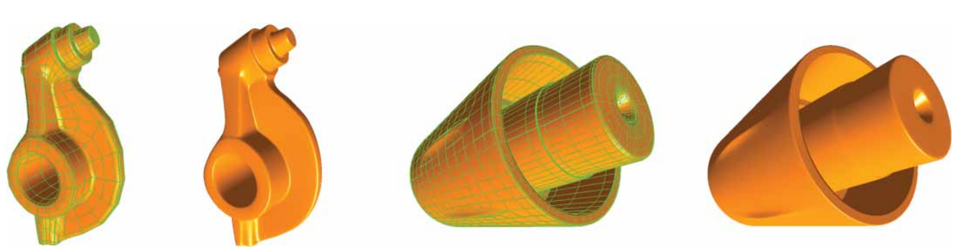

Fig. 5: Two examples of constructed interpolating surfaces with Bezier Crust on CCSS.

Δp_3 are the offsetting difference vectors at four corners of a CCS face, as shown in Fig. 7.

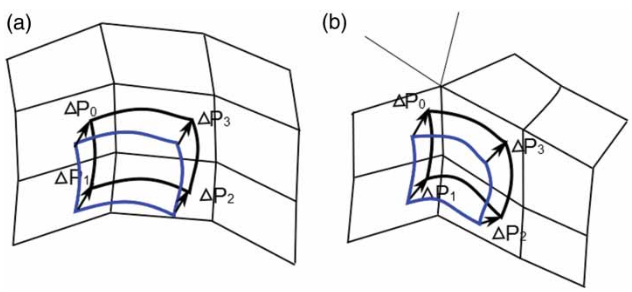

Fig. 7: offsetting surface (blue) obtained after subtracting offsetting Bezier crust from the CCS limit surface: (a) regular face (b) extraordinary face.

With offsetting Bezier crust $\Delta S(u, v)$ defined, the offsetting parametric surface $\bar{S}(u, v)$ can be expressed as follows:

$$\bar{S}(u, v) = S(u, v) - \Delta S(u, v) \tag{3.3}$$

In the above, the construction of an offsetting parametric surface for a CCSS is shown. Next, we will analyze the behavior of this new scheme and show some properties of this offsetting surface.

4. BEHAVIOR OF THE NEW OFFSETTING SURFACE AND DISCUSSION

In this section, we discuss the behavior of our new offsetting surface.

THEOREM 1: the new offsetting parametric surface $\bar{S}(u, v)$ is C^2 continuous everywhere, except at extraordinary points, where it is C^1 continuous.

PROOF: Our new offsetting parametric surface is constructed by subtracting an offsetting Bezier crust from the CCS limit surface. CCS limit surface is C^2 continuous everywhere, except at extraordinary points, where it is C^1 continuous. And by Wang and Cheng [15], the bi-quintic Bezier crust is C^2 continuous everywhere except at corner control points and across the face boundaries, where its derivatives vanish up to the 2nd order. Computing 1st and 2nd order derivatives on (3.3), it will show that $\bar{S}(u, v)$ will maintain the curvature continuity of CCS limit surface $S(u, v)$. QED

One of the benefits to subtract an offsetting Bezier crust from CCS limit surface is that the offsetting Bezier crust has the same behavior to handle both regular face and extraordinary face. This derives from the fact that, given the required thickness of d, the offsetting Bezier crust works on difference vectors of size d and with the direction of the unit normal of corner data points. Given a regular or extraordinary face of degree N, computation of the offsetting Bezier crust is independent of N.

With Eigen-decomposition, each individual data point on the CCS limit surface can be computed in $O(1)$. Calculating an arbitrary point on offsetting Bezier crust by (3.2) is also $O(1)$. Such that computation of each individual limit point on offsetting surface is $O(1)$. It is apparently more efficient in comparison with constructing offsetting surface layer by layer by slicing.

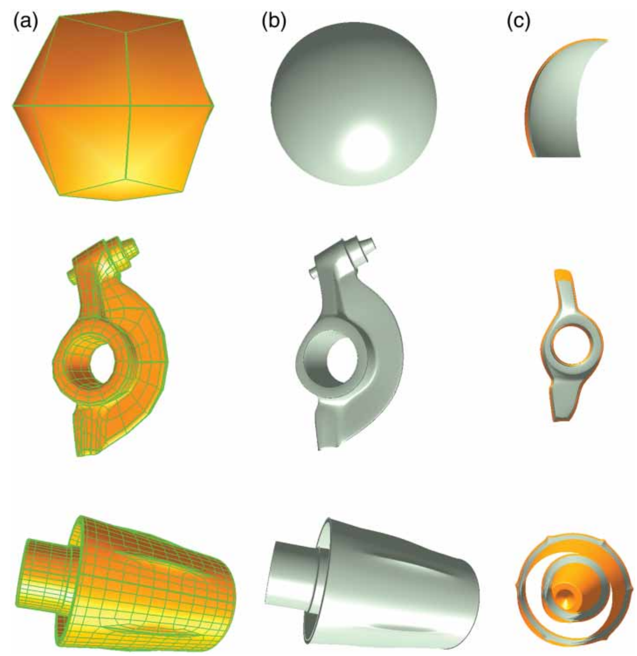

Fig. 8: Three examples: (a) CCS control mesh; (b) new offsetting surface (without the CCS surface); (c) cross-sectional view - yellow is outer CCS surface, gray is the offsetting surface.

Given a CCS face, when it is flat, the difference vectors on all four corners are equal, with (3.3), each limit point on obtained offsetting face has exactly the same geodesic distance d to original limit surface. When the face is concave or convex, then a limit point of $\bar{S}(u, v)$ on parametric surface is the sum of the limit point $S(u, v)$ and affine combination of four difference vectors $\Delta p_i = d \cdot \vec{n_i}, i = 1, 2, 3, 4$. We derive that

$$|\Delta S(u, v)| \leq |d \cdot \vec{n}_{S(u,v)}| \tag{4.1}$$

where $|\cdot|$ is the size of the enclosed vector.

With (4.1), we can further derive that

$$\|\bar{S}(u, v) - S(u', v')\| = d - \varepsilon, \tag{4.2}$$

where $\|\ .\ \|$ represents the shortest distance between the offsetting surface $\acute{S}$ and the original CCS surface S. ε is the maximum error.

Further, we can also show that our new offsetting surface is enclosed in the original CCS surface and the conventional offsetting surface defined in (3.1).

Since $\Delta S(u, v)$ is affine combination of difference vectors at 4 corners. To reduce ε, we can perform more CCS subdivisions on original CCS control mesh. If we define M^n as the CCS control mesh after nth subdivision, we can derive that

$$\text{when } n \to \infty, |\Delta S(u, v)| \approx |d \cdot \vec{n}_{S(u,v)}| \tag{4.3}$$

which is exactly the representation of the conventional way of surface offsetting shown in (2.4) with $\varepsilon \approx 0$. Note the surface generated by (2.4) is generally not a smooth surface. Fig. 9 shows how the subdivisions impact the surface quality of the offsetting surface. In the center where curvature is high, the offsetting surface shows increasing creases after three times subdivision, whereas original one is smooth. This is consistent with our analysis shown in Fig. 3 of section 2.

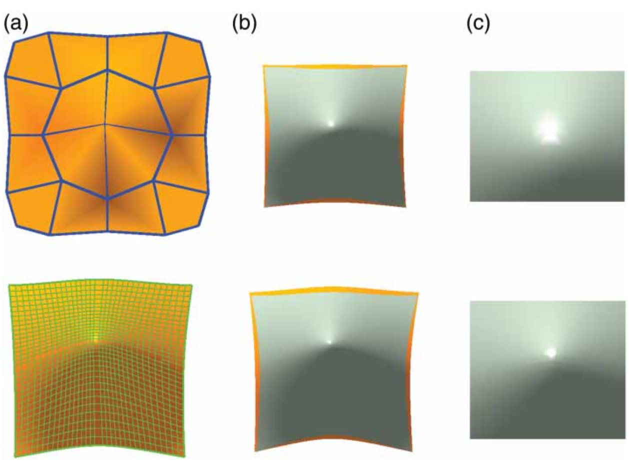

Fig. 9: Comparison of offsetting surface after 3 recursive subdivisions on CCS mesh: (a) CCS control mesh, (b) generated offsetting surface, (c) enlarged central part of the offsetting surface.

Since in general cases ε is small and we do want to avoid the offsetting surface obtained from (2.4) (many creases and self-intersection when outer surface is convex), so it will not be necessary to perform further subdivision if ε is within the tolerance.

Figure 9: Comparison of offsetting surface after 3 recursive subdivisions on CCS mesh: (a) CCS control mesh, (b) generated offsetting surface, (c) enlarged central part of the offsetting surface.

Our scheme is based on the assumption that all corner CCS data points have non-zero unit normal, we will also include discussion of scenario when unit normal does not exist (control mesh collapses). Prerequisite of (3.3) is that on each corner data point of the CCS limit surface its unit normal exists. In most cases, it is true, however there are some special cases where unit normal does not exist (1^{st} order derivative along one parametric direction is 0, due to control vertices coinciding). In such rare cases, we propose to add the unit normal to such corner data point with the average of the unit normal's on its neighboring data point. The algorithm is as follows,

a. If all CCS corner data points have unit normal then go to (c), otherwise pick up a data point where unit normal does not exist, go to (b).
b. For this data point, we put average of its neighboring unit normal as its unit normal. Go back to (a).
c. End of the algorithm, start to construct the offsetting surfaces.

Above algorithm is heuristic, since it defines the unit normal on some collapsed control vertex as the average of its neighboring unit normal's when its unit normal does not exist. Further research needs to be made to handle such special cases.

Implementation results in Fig. 1 and Fig. 8 show that a smooth thin offsetting surface can be generated by applying our new scheme. The offsetting surface keeps a quasi-uniform thickness with CCS limit surfaces, which formed a nice hollowed 3D solid appropriate for common CAD usage.

5. CONCLUSION

In this paper, we introduce a new thin shell hollowing model on 3D objects represented by Catmull-Clark subdivision surfaces. Our new method for inward offsetting works by subtracting a thin layer of bi-quintic Bezier crust from the original CCS surface.

The new offsetting surface generated is visually smooth and has the same continuity as the original CCS limit surface, i.e. C^2 continuous everywhere, except at extraordinary points, where it is C^1 continuous. The properties of new offsetting surface are also discussed in this paper.

Implementation results show that the offsetting surface generated is free from creases, and filling the gaps is trivial due to the fact that the offsetting surface is the sum of the original CCS parametric surface and a Bezier crust on difference vectors of size d on each face. Since a bi-quintic Bezier crust does not change the curvature at a corner data point of the CCS limit surface, one would not get gaps at connections of offsetting faces commonly found in earlier methods.

Our next step is to explore current solutions of removing unwanted loops, and apply them to our new scheme to generate a smooth 3D offsetting surface without creases, loops and self-intersections.

ACKNOWLEDGMENT

This work is supported by National Science Foundation of China (61020106001, 61170324), National Science Council of ROC (NSC-100-2811-E-007-021), and a joint grant of National Tsinghua University and Chang-Gung Memorial Hospital (101N2756E1).

REFERENCES

[1] Catmull, E.; Clark, J.: Recursively generated B-spline surfaces on arbitrary topological meshes, Computer-Aided Design, 10(6), 1978, 350–355. http://dx.doi.org/10.1016/0010-4485(78)90110-0

[2] Chiu, W.; Tan, S.: Using dexels to make hollow models for rapid prototyping, Computer-Aided Design, 30(7), 1998, 539–547. http://dx.doi.org/10.1016/S0010-4485(98)00008-6

[3] Elber, G.; Cohen, E.: Adaptive isocurve-based rendering for freeform surfaces, ACM Transactions on Graphics, 15(3), 1996, 249–263. http://dx.doi.org/10.1145/231731.231736

[4] Elber, G.; Cohen, E.: Tool path generation for freeform surface models, In Proceedings on the second ACM symposium on Solid modeling and applications, 1993, 419–428. http://dx.doi.org/10.1145/164360.164500

[5] Ganesan, M.; Fadel, G. M.: Hollowing rapid prototyping parts using offsetting techniques, In Proceedings of the Fifth International Conference on Rapid Prototyping, 1994, 241–251, University of Dayton.

[6] Hatna, A.; Grieve R.; Broomhead, P.: Offsetting 3D contours on parametric surfaces, The International Journal of Advanced Manufacturing Technology, 16(3), 2000, 189–195. http://dx.doi.org/10.1007/s001700050026

[7] Kahmann, J.: Continuity of curvature between adjacent Bezier patches, Surfaces in CAGD, North-Holland, Amsterdam, 65–75, 1983.

[8] Koc, B.; Lee Y.-S.: Non-uniform offsetting and hollowing objects by using biarcs fitting for rapid prototyping processes, Computers in Industry, 47(1), 2002, 1–23. http://dx.doi.org/10.1016/S0166-3615(01)00141-5

[9] Lai, S.; Cheng, F.: Parametrization of general Catmull-Clark subdivision surfaces and its applications, Computer-Aided Design and Applications, 3(1–4), 2006, 513–522. http://dx.doi.org/10.1080/16864360.2006.10738490

[10] Lam, T.; Yu, K.; Cheung, K. et al.: Octree reinforced thin-shell rapid prototyping, Journal of materials processing technology, 63(1), 1997, 784–787. http://dx.doi.org/10.1016/S0924-0136(96)02724-0

[11] Li, C.; Yu, K.; Lam, T.: Implementation and evaluation of thin-shell rapid prototype, Computers in Industry, 35(2), 1998, 185–193. http://dx.doi.org/10.1016/S0166-3615(97)00049-3

[12] Maekawa, T.: An overview of offset curves and surfaces, Computer-Aided Design, 31(3), 1999, 165–173. http://dx.doi.org/10.1016/S0010-4485(99)00013-5

[13] Park, S. C.: Hollowing objects with uniform wall thickness, Computer-Aided Design, 37(4), 2005, 451–460. http://dx.doi.org/10.1016/j.cad.2004.08.001

[14] Patrikalakis, N.; Bardis, L.: Offsets of curves on rational B-spline surfaces, Engineering with Computers, 5(1), 1989, 39–46. http://dx.doi.org/10.1007/BF01201996

[15] Wang, J.; Cheng, F.: Bezier Crust on Quad Subdivision Surface, In Pacific Graphics Short Papers, 2013, 29–34, The Eurographics Association.