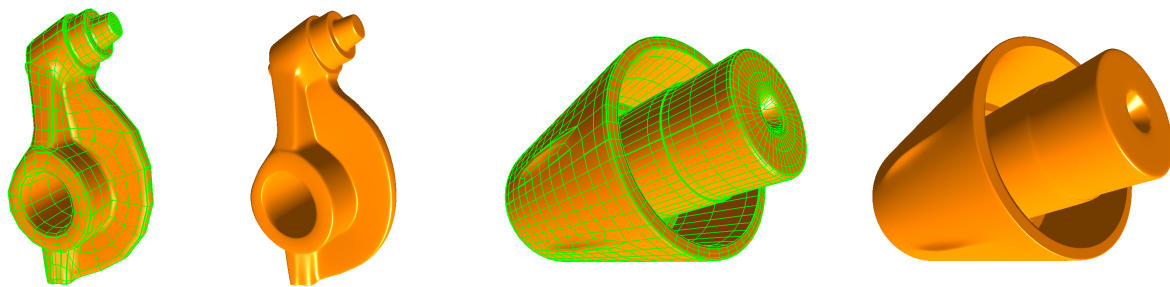


# Bezier Crust on Quad Subdivision Surface

Jianzhong Wang<sup>1</sup> and Fuhua Cheng<sup>1</sup>

<sup>1</sup>Computer Science Department, University of Kentucky, USA



**Figure 1:** Two examples of Bezier crust applied on Catmull-Clark subdivision surfaces

## Abstract

Subdivision surfaces have been widely used in computer graphics and can be classified into two categories, approximating and interpolatory. Representative approximating schemes are Catmull-Clark (quad) and Loop (triangular). Although widely used, one issue remains with the approximating schemes, i.e., the process of interpolating a set of data points is a global process so it is difficult to interpolate large data sets. In this paper, we present a local interpolation scheme for quad subdivision surfaces through appending a  $G^2$  Bezier crust to the underlying surface, and show that this local interpolation scheme does not change the curvatures across the boundaries of underlying subdivision patches, therefore, one obtains high quality interpolating limit surfaces for engineering and graphics applications efficiently.

Categories and Subject Descriptors (according to ACM CCS): I.3.5 [Computer Graphics]: Computational Geometry and Object Modeling—Curve, surface, solid, and object representations

## 1. Introduction

Subdivision surfaces have been widely used in surface representation. Compared to traditional spline methods (e.g. Bezier Surface), advantages include simpler to use and can work on any topology.

Subdivision schemes use three types of mesh structure: quadrilateral, triangular and hexagonal. Quad faces and Triangular faces are most commonly used for practical applications. Subdivision surfaces can be classified into two types: face-split and vertex-split. Vertex-split schemes (midedge [DS78], biquartic [Qu90]) are not as popular as face-split schemes because they do not generate well behaved surfaces on an arbitrary topology as face-split schemes. In a

face-split scheme, vertices of the control mesh are refined recursively. Each vertex of the current control mesh is redefined in the next subdivision level. If the original vertex and its corresponding vertex in the next subdivision step are the same, we call this scheme interpolating (e.g. Modified Butterfly [DLG90], Kobbelt [Kob96]), otherwise the scheme is approximating (e.g. Loop [Loo87], Catmull-Clark [CC78]). Interpolating is attractive, since vertices in the original control mesh remain in the control meshes in subsequent subdivisions, making subdivision more intuitive. However, surface quality of interpolating schemes is not as good as that of approximating schemes. As a comparison, interpolating schemes such as Modified Butterfly and Kobbelt scheme

are  $C^1$  continuous on regular meshes, while approximating schemes such as Catmull-Clark and Loop are  $C^2$  continuous on regular meshes. Among various subdivision schemes, Loop and Catmull-Clark are most widely used on triangular meshes and quad meshes, respectively.

As an approximating scheme, limit surface of the Loop subdivision or the Catmull-Clark subdivision (CCS) does not interpolate the control mesh in general. However, since construction of smooth interpolating surfaces is important in many applications, including CAD, statistical data modeling and face recognition, it is necessary to develop interpolation techniques for approximating subdivision schemes. In this paper, we will address the issue of interpolating quad data meshes, focusing especially on Catmull-Clark scheme.

Given a quad data mesh, the process of calculating a CCS control mesh whose limit surface interpolates the given data mesh can be done directly or iteratively. A direct method such as the earlier work of Halstead [HKD93] is not recommended because calculating the inverse of a large matrix is not feasible (the number of data points in an interpolation problem is typically hundreds or even thousands). Iterative methods, on the other hand, do not need to compute the inverse of a large matrix [BT09] [CLT\*08]. Some of them even have an exponential convergence rate [CLT\*08]. But the interpolation is basically an approximating process.

In this paper, we present a simple interpolation scheme for CCS. The new scheme interpolates the given data mesh precisely, instead of iteratively. It works by appending the parametric polynomial of a special bi-quintic Bezier crust to a Catmull-Clark parametric surface. The Bezier crust works on difference vectors between CCS control points and corresponding data points, so the new interpolating surface can be computed locally. There is no need to solve a global linear system and the algorithm is efficient and compact. With special properties of Bezier crust at the boundaries of a surface patch, the new interpolation scheme works on an arbitrary quad subdivision surface as well, and will maintain its  $C^1/C^2$  continuity.

The rest of paper is organized as follows: section 2 reviews previous approaches of mesh interpolation, section 3 presents the concept of Bezier crust on space curve, section 4 introduces the new interpolating parametric surface by appending tensor-product Bezier crusts to a quad subdivision surface with focus on Catmull-Clark, section 5 shows several implementations and a discussion, section 6 concludes.

## 2. Previous Works

In this section, we briefly review earlier methods for interpolation of given data meshes by quad subdivision scheme of Catmull-Clark and by traditional spline scheme of Bezier surface. The goal of the interpolation is to get a smooth limit surface that is tangent plane continuous ( $G^1$ ) or curvature continuous ( $G^2$ ). In this paper, we focus on  $G^2$  surfaces,

which are suitable for most engineering and graphics applications.

### 2.1. Interpolating Scheme of Catmull-Clark

Catmull-Clark subdivision (CCS) is the most widely used subdivision scheme. Control points in a CCS control mesh can be classified into three categories: vertex, edge and face [CC78]. In each CCS, a new face point is created for each face, a new edge point is created for each edge, and the original vertex points are updated with new vertex points. By performing recursive subdivision, one can obtain a limit surface that is  $C^2$  everywhere except at extraordinary points, where it is  $C^1$  (tangent plane) continuous only [BS88] [DS78].

Interpolation with a CCS surface can be performed by solving a linear system,

$$\mathbf{A}x = b \quad (1)$$

where  $\mathbf{A}$  is a square matrix determined by interpolation conditions and mesh topology,  $x$  is a column vector of control points to be determined,  $b$  is a column vector of data points in the given data mesh [HKD93]. If  $\mathbf{A}$  is a small and non-singular matrix, we can obtain the control mesh by calculating  $\mathbf{A}^{-1}$  directly first. However, a direct method will not work or not work well if  $\mathbf{A}$  is a singular or large matrix. In such a case, an iterative method needs to be applied. Traditionally, stationary iterative methods like Jacobi, Gauss-Seidel or Successive Over-relaxation can be used to solve a large linear system. The issue with these methods is the convergence rate - they are slow when the data set is large. When  $\mathbf{A}$  is singular, the least-squares method can be applied. There are faster iterative methods to solve larger scale data sets [BT09] [Sze90]. However, since (1) is a global system, convergence rate will still not be satisfactory when we are dealing with thousands of data points.

To avoid dealing with singular linear systems and to improve iteration speed, a progressive subdivision scheme [CLT\*08] [CFL\*09] has been developed. This method iteratively generates a new control mesh by adding to old control mesh the difference between this control mesh and its corresponding data points on the CCS limit surface and shows that the linear system developed is positive definite and can improve the convergence speed of CCS control mesh generation process which satisfies (1).

Besides convergence speed, the interpolating surface obtained by solving (1) sometime is unsatisfactory because of excessive undulations [HKD93]. Halstead [HKD93] notices that the undulations appear because they are not indicated by the shape of the original mesh. The Fairing techniques proposed in [LP88] [ZZC01] smooth an interpolating surface by including more constraints but increasing the size of the control mesh. Some alternative methods [LC06] [ZC06] improve shapes by choosing good initial control mesh or adding more control points to control the shape locally.

The above methods focus on improving convergence speed of solving (1) or introducing additional constraints to handle surface artifact, they are all approximating schemes. It is natural to ask the following question: "Is it possible to have a precise interpolating scheme other than approximating ones, without solving a global linear system, but not iterative, while preserving the easy implementation and curvature continuity features of CCS?"

## 2.2. $G^2$ Bezier Surface

In CAGD, Bezier Patch is one of the most widely used representations in free-form surface modeling. Since each Bezier Patch interpolates its 4 corner control points, this makes it a natural choice in surface construction when an interpolating scheme is desired.

A two-dimensional Bezier surface patch can be defined as a parametric surface,

$$\mathbf{p}(u, v) = \sum_{i=0}^n \sum_{j=0}^m \mathbf{b}_{i,n}(u) \mathbf{b}_{j,m}(v) \mathbf{P}_{i,j}, \quad (2)$$

where  $\mathbf{b}_{i,n}(u)$  and  $\mathbf{b}_{j,m}(v)$  are Bernstein basis functions of degree  $m$  and  $n$ , respectively, and  $\mathbf{P}_{i,j}$  are control points. Since the commonly used Bezier patch has  $m = n$ , so here we focus on piecewise tensor-product Bezier surface only.

It is clear from the definition (2) that the four corner control points are interpolated by its limit surface. Conditions of  $G^1$  continuity for a piecewise Bezier surface were discussed in [Bez86] [DeR90] [LH89]. It was pointed out, to obtain  $G^1$  continuity, one must ensure that partial derivatives across the boundary of Bezier patches ( $n \geq 2$ ) must be coplanar.

In CAGD,  $G^2$  continuity is necessary to ensure the existence of a visually well behaved surface. Conditions for  $G^2$  continuity are discussed in [Deg90] [Kah83] [YLN96]. These works show that, to get  $G^2$  continuity, one must have at least a piecewise biquintic Bezier surface. Although one can theoretically obtain a piecewise  $G^2$  Bezier surface, the construction of such a surface is more difficult than the construction of a subdivision surface. One has to solve a linear system of partial derivatives up to second order across the boundaries, and the linear system has too many degrees of freedom. Gregory reduces the degree of freedom by introducing constraints on internal control points of a Bezier patch [GH89], but its construction is still not an easy task.

In the above we have reviewed two main interpolating schemes: subdivision surface based and Bezier patch based. The first scheme is approximating and suffers problems with convergence speed and undesired undulations, while the latter one is more difficult to construct.

## 3. Bezier Crust on Space Curve

In this section, we introduce a special quintic Bezier off-set polynomial named Bezier crust on curve. We show that

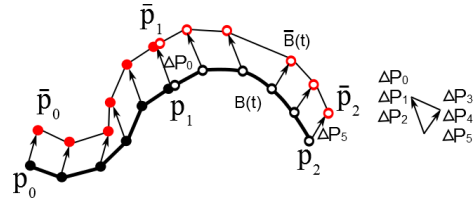
when a Bezier crust is added to a  $C^2$  space curve, the new curve is  $C^2$  with the same tangent and curvature at the start and end points.

A Bezier spline is a composite curve formed by piecing together several Bezier curve segments. A Bezier spline interpolates all the start and end control points of its Bezier curve segments. While quadratic and cubic Bezier splines are widely used in font design and 3D animation, they are not  $G^2$  continuous between adjacent Bezier segments. To obtain a  $G^2$  Bezier spline, quintic Bezier curve segments are needed [Deg90].

A quintic Bezier curve segment takes the following form,

$$\mathbf{B}(t) = \sum_{i=0}^5 \mathbf{b}_{i,5}(t) \mathbf{P}_i, \quad (3)$$

where  $\mathbf{b}_{i,5}(t) = \binom{5}{i} t^i (1-t)^{5-i}$ ,  $i = 0, \dots, 5$ , are Bernstein polynomials of degree 5 and  $\mathbf{P}_i$  are its control point.



**Figure 2:** Control points of  $\bar{\mathbf{B}}(t)$  after movement of  $\Delta \mathbf{P}(t)$  from original  $\mathbf{B}(t)$ .

Fig 2 shows the movement of control points of a quintic Bezier spline composed of two Bezier curve segments. The Bezier spline interpolates  $P_0$ ,  $P_1$  and  $P_2$ . We want it to interpolate  $\bar{P}_0$ ,  $\bar{P}_1$ , and  $\bar{P}_2$ , here we only consider the six control points of the right Bezier curve segment  $\mathbf{B}(t)$ , the left curve segment can be adjusted similarly. If we want to maintain its curvature at the start and end points on the new Bezier curve segment  $\bar{\mathbf{B}}(t)$ , we can set  $\bar{\mathbf{B}}(t)$  as

$$\bar{\mathbf{B}}(t) = \mathbf{B}(t) + \sum_{i=0}^5 \mathbf{b}_{i,5}(t) \Delta \mathbf{P}_i, \quad (4)$$

where  $\Delta \mathbf{P}_0 = \Delta \mathbf{P}_1 = \Delta \mathbf{P}_2 = \bar{p}_1 - p_1$  and  $\Delta \mathbf{P}_3 = \Delta \mathbf{P}_4 = \Delta \mathbf{P}_5 = \bar{p}_2 - p_2$ .

If we define

$$\Delta \mathbf{B}(t) = \sum_{i=0}^5 \mathbf{b}_{i,5}(t) \Delta \mathbf{P}_i, \quad (5)$$

we have the following properties on  $\Delta \mathbf{B}(t)$  of degree 5:

- (1) when  $\Delta \mathbf{B}(t)$  is displayed alone, it is a line segment independent of its degree. So it is  $C^2$  on  $\Delta \mathbf{B}(t)$  except at the start and ending points.

- (2) The 1<sup>st</sup> and 2<sup>nd</sup> derivatives of  $\Delta\mathbf{B}(t)$  at the start and end points are both zero. So it will not change the 1<sup>st</sup> and 2<sup>nd</sup> derivatives of  $\mathbf{B}(t)$  at the start and end points.
- (3) The new quintic Bezier spline obtained by adding  $\Delta\mathbf{B}(t)$  to each of the original Bezier curve segments will remain  $G^2$  continuous if the original Bezier spline is  $G^2$ .

With the above properties, we name  $\Delta\mathbf{B}(t)$  of degree 5 a Quintic Bezier crust. We notice that this Quintic Bezier crust can be added to an arbitrary  $C^2$  curve.

**THEOREM 1:** The new curve obtained by adding a Quintic Bezier crust to a  $C^2$  parametric curve is  $C^2$  continuous and has the same curvature at the start and end point as the original curve.

**PROOF:** A  $C^2$  parametric curve can be written in polynomial form at a parametric value  $t_0$  as

$$f(t) = f(t_0) + f'(t_0)(t - t_0) + \frac{f''(t_0)}{2}(t - t_0)^2 + \delta. \quad (6)$$

The new curve  $\tilde{f}(t) = f(t) + \Delta\mathbf{B}(t)$ , by calculating its first and second derivatives, we can prove that  $\tilde{f}(t)$  is  $C^2$  and has the same curvature at the start and end points as  $f(t)$ . QED

Since a piecewise cubic B-spline curve is  $C^2$ , we can add quintic Bezier crust (with difference vectors chosen as differences between control points and their corresponding data points) to each curve segment and obtain a new  $C^2$  composite curve which interpolates all control points of the original curve (except 1st and last control points if the curve is open).

#### 4. Bezier Crust on Quad Subdivision Surface

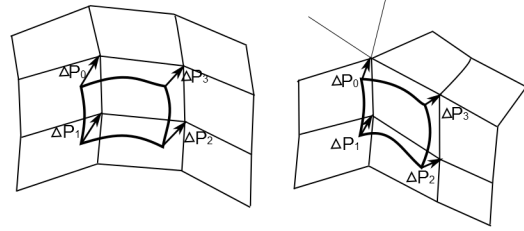
In this section, we introduce a new interpolating scheme for quad subdivision surfaces like Catmull-Clark. The new scheme will interpolate a given data mesh exactly.

Quad subdivision schemes have been widely used in surface representation because of their simplicity and well behaved limit surfaces. Among various quad schemes, Doo Sabin [DS78], Mid-Edge [PR97] are  $C^1$  continuous, Catmull-Clark [CC78] is  $C^2$  everywhere except at extraordinary points. In this paper, we present a new unified interpolating scheme for quad approximating subdivision surfaces, with main effort focusing on Catmull-Clark.

Given a quad control mesh  $\mathbf{M}$ , the CCS scheme generates a limit surface that approximates the control mesh. The limit surface of each face  $f$  of  $\mathbf{M}$  (regular or extraordinary) can be represented in parametric form  $S(u, v)$ . For each  $f$ , we define  $\Delta\mathbf{P}_0$ ,  $\Delta\mathbf{P}_1$ ,  $\Delta\mathbf{P}_2$  and  $\Delta\mathbf{P}_3$  (Fig 3) as the difference vectors between its corner control points and its corresponding data points, respectively. In order to interpolate the corner control points, similar to quintic Bezier crust, we can define a bi-quintic Bezier crust  $\Delta\mathbf{p}(u, v)$  as follows,

$$\Delta\mathbf{p}(u, v) = \sum_{i=0}^5 \sum_{j=0}^5 \mathbf{b}_{i,5}(u) \mathbf{b}_{j,5}(v) \Delta\mathbf{P}_{i,j}, \quad (7)$$

where  $\Delta\mathbf{P}_{i,j}$  are control points of a bi-quintic Bezier surface, and  $\Delta\mathbf{P}_{i,j} = \Delta\mathbf{P}_0$  if  $i \in [0, 2]$  &  $j \in [0, 2]$ ,  $\Delta\mathbf{P}_{i,j} = \Delta\mathbf{P}_1$  if  $i \in [0, 2]$  &  $j \in [3, 5]$ ,  $\Delta\mathbf{P}_{i,j} = \Delta\mathbf{P}_2$  if  $i \in [3, 5]$  &  $j \in [0, 2]$ ,  $\Delta\mathbf{P}_{i,j} = \Delta\mathbf{P}_3$  if  $i \in [3, 5]$  &  $j \in [3, 5]$ .



**Figure 3:** Difference vectors between control points and limit points of a regular(left)/extraordinary(right) CCS face

When displayed by itself, the Bezier crust defined in (7) has exactly the same boundaries as a bilinear Coons patch. By analyzing the 1<sup>st</sup> and 2<sup>nd</sup> order derivatives, we get the following properties of a bi-quintic Bezier crust:

- (1) At the four corners, the 1<sup>st</sup> and 2<sup>nd</sup> order derivatives vanish.
- (2) At the four boundaries, the 1<sup>st</sup> and 2<sup>nd</sup> order derivatives across the boundaries vanish. Since the difference vectors along the boundary are the same for neighboring Bezier crusts, the boundary curve between neighboring Bezier crusts coincides.
- (3) At  $(u, v)$  of the Bezier crust, the 1<sup>st</sup> and 2<sup>nd</sup> order derivatives are continuous or vanishes.
- (4) Bezier crust works on difference vectors at the four corners of a surface patch, so it has the same representation form for both regular and extraordinary face.

By adding the Catmull-Clark parametric form  $S(u, v)$  to its Bezier crust  $\Delta\mathbf{p}(u, v)$ , we obtain a parametric surface  $\tilde{S}(u, v)$  which interpolates the four corner control points of  $f$ , as follows:

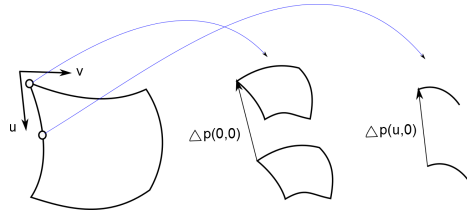
$$\tilde{S}(u, v) = S(u, v) + \Delta\mathbf{p}(u, v) \quad (8)$$

$S(u, v)$  is computed locally with its  $(2N + 8)$  control points ( $N$  is the valence). Since the difference vectors can be calculated locally with its surrounding  $(2N + 1)$  control points,  $\Delta\mathbf{p}(u, v)$  is also computed locally. So (8) differs from earlier CCS interpolation schemes in that it is a local piecewise parametric surface. Hence, it is not necessary to calculate directly or iteratively a global new control mesh to interpolate a given control mesh (shown in (1)).

**THEOREM 2:** The limit surface of an interpolating surface patch  $\tilde{S}(u, v)$  defined in (8) is  $C^2$  continuous everywhere except at extraordinary points.

**PROOF:** By properties (1) and (2), the first and second order derivatives at four corners and across the boundaries of a Bezier crust vanish, so the continuity remain the same at the corners and across the boundaries for both regular

and extraordinary faces. By properties (2) and (3), we can also show  $C^2$  continuity along the boundaries and on the new surface patch. So the limit surface of our new scheme maintains  $C^2$  continuity everywhere except at extraordinary points where it is  $C^1$  continuous. QED



**Figure 4:** Behavior at corner limit points (middle) and across-boundary limit points (right) after adding Bezier crust to CCS (left)

Since a Bezier crust depends only on the difference vectors, we can apply Bezier crusts to any quad approximating scheme, as far as the subdivision surface can be parameterized. The following is a generalization of Bezier crusts on quad subdivision surface. A quad subdivision surfaces is either  $C^1$  or  $C^2$ .

**PROPOSITION 1:** Bi-quintic Bezier crusts can be added to any parametric  $C^1$  or  $C^2$  quad subdivision surface. The new parametric surface interpolates the given control mesh, while maintains the continuity of the original subdivision surface.

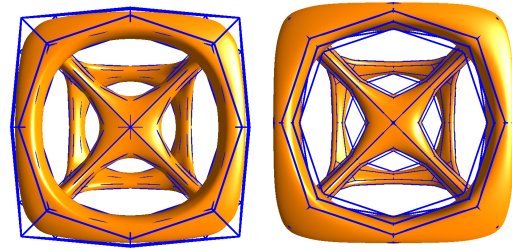
**PROOF:** Difference vectors between control points and corresponding limit points of the original subdivision surface are determined by its subdivision rules. By adding Bezier crusts based on these difference vectors to the given subdivision surface, the new parametric surface interpolates the given data mesh and keeps the continuity of the original subdivision surface by properties (1), (2) and (3) of a bi-quintic Bezier crust. QED

In the above, we have showed the construction of a new parametric interpolating surface by adding bi-quintic Bezier crusts to a Catmull-Clark subdivision surface, and presented a general solution to arbitrary parametric quad subdivision schemes. Next, we will show implementation and analysis of this new parametric surface.

**5. Implementation and Discussion**

In last sections, we introduce a new interpolation scheme for quad subdivision surfaces with a focus on Catmull-Clark. Here we review the implementation and analyze the Bezier crust on CCS.

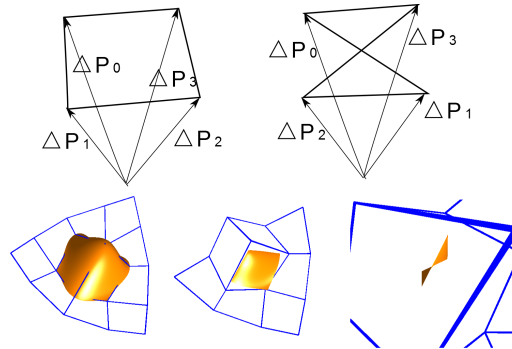
As shown in (7) and (8), our new interpolating scheme on CCS is obtained locally by adding the parametric polynomial of a local Bezier crust to a CCS surface patch. It is far



**Figure 5:** Left side shows a CCS limit surface, right side shows the limit surface of our new interpolating scheme

superiorer than the earlier global interpolation schemes. The algorithm is numerically stable and compact. Fig. 1 shows two engineering parts with our new Bezier crust method on CCS. Fig. 5 shows a comparison of limit surfaces on a given control mesh by the original CCS and by our interpolating scheme. From the images, one can see that the limit surface of our new scheme is well behaved.

Since difference vectors between CCS control points and data points can be of any values, a Bezier crust could be quite normal in one case while not so normal in another case. For instance, in Fig. 6, top left side shows a normal patch, while on the top right side the patch is twisted. Obviously, for a standalone surface a twisted control mesh is undesired for it will render the surface patch not well behaved. Nevertheless, with our interpolating scheme, since the Bezier crust is added to the underlying CCS patch, the twisting effect can be offset by the underlying CCS patch, such that the new surface patch still maintains its continuity (Fig 6).



**Figure 6:** Top row shows two scenario of Bezier crust, bottom row shows a twisted example. In the bottom row, limit surfaces of new interpolating scheme (left), a CCS (middle) patch and its standalone Bezier crust (right, enlarged)

One limitation we notice is that, when a Bezier crust is displayed by itself,  $\Delta \mathbf{p}(u, v)$  is enclosed in the volume bounded by  $\Delta \mathbf{P}_0, \Delta \mathbf{P}_1, \Delta \mathbf{P}_2$  and  $\Delta \mathbf{P}_3$ , so the generated interpolating limit surface might show diminishing effect on



curvature towards the center. From Fig. 5, we can see several slightly flattened surface areas. More experiments need to be done to see if this will cause any unwanted surface artifacts.

## 6. Conclusion

In this paper, we introduce a simple interpolation scheme for quad parametric subdivision surfaces. We show that by adding a special bi-quintic Bezier crust to each of the original subdivision surface patches one can generate an interpolating surface that maintains curvature conditions of the original limit surfaces.

With a special construction of bi-quintic Bezier crusts, we can avoid the calculation of a global linear system common in earlier interpolation schemes, but get a system that is local and simple. Implementation results on CCSS show that the new interpolating scheme can generate visually well behaved limit surfaces, such that barely no fairing is needed.

Our Bezier crust interpolating scheme is limited to quad subdivision surfaces. For triangular subdivision surfaces (e.g. Loop) which are also popular in computer graphics, different schemes have to be developed. That will be one of our future works.

In summary, we provide a local interpolating scheme for quad subdivision surfaces. With the simplicity of this scheme, one can easily apply it to approximating subdivision surfaces, making them more appropriate for CAD, CAGD, face recognition and other interpolation-demanding applications.

## 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

- [Bez86] BEZIER P.: *The mathematical basis of the UNISURF CAD system*. Butterworth-Heinemann, 1986. 3
- [BS88] BALL A., STORRY D.: Conditions for tangent plane continuity over recursively generated B-spline surfaces. *ACM Transactions on Graphics (TOG)* 7, 2 (1988), 102. 2
- [BT09] BECK A., TEBoulLE M.: A fast iterative shrinkage-thresholding algorithm for linear inverse problems. *SIAM Journal on Imaging Sciences* 2, 1 (2009), 183–202. 2
- [CC78] CATMULL E., CLARK J.: Recursively generated B-spline surfaces on arbitrary topological meshes. *Computer-Aided Design* 10, 6 (1978), 350–355. 1, 2, 4
- [CFL\*09] CHENG F.-H. F., FAN F.-T., LAI S.-H., HUANG C.-L., WANG J.-X., YONG J.-H.: Loop subdivision surface based progressive interpolation. *Journal of Computer Science and Technology* 24, 1 (2009), 39–46. 2
- [CLT\*08] CHEN Z., LUO X., TAN L., YE B., CHEN J.: Progressive interpolation based on catmull-clark subdivision surfaces. In *Computer Graphics Forum* (2008), vol. 27, Wiley Online Library, pp. 1823–1827. 2
- [Deg90] DEGEN W. L.: Explicit continuity conditions for adjacent bézier surface patches. *Computer Aided Geometric Design* 7, 1 (1990), 181–189. 3
- [DeR90] DEROSE T. D.: Necessary and sufficient conditions for tangent plane continuity of bezier surfaces. *Computer Aided Geometric Design* 7, 1 (1990), 165–179. 3
- [DLG90] DYN N., LEVINE D., GREGORY J. A.: A butterfly subdivision scheme for surface interpolation with tension control. *ACM transactions on Graphics (TOG)* 9, 2 (1990), 160–169. 1
- [DS78] DOO D., SABIN M.: Behaviour of recursive division surfaces near extraordinary points. *Computer-Aided Design* 10, 6 (1978), 356–360. 1, 2, 4
- [GH89] GREGORY J. A., HAHN J. M.: A c2 polygonal surface patch. *Computer Aided Geometric Design* 6, 1 (1989), 69–75. 3
- [HKD93] HALSTEAD M., KASS M., DEROSE T.: Efficient, fair interpolation using Catmull-Clark surfaces. In *Proceedings of the 20th annual conference on Computer graphics and interactive techniques* (1993), ACM, p. 44. 2
- [Kah83] KAHMANN J.: Continuity of curvature between adjacent bézier patches. *Surfaces in Computer Aided Geometric Design* (1983), 65–75. 3
- [Kob96] KOBBELT L.: Interpolatory subdivision on open quadrilateral nets with arbitrary topology. In *Computer Graphics Forum* (1996), vol. 15, John Wiley & Sons, pp. 409–420. 1
- [LC06] LAI S., CHENG F.: Similarity based interpolation using Catmull-Clark subdivision surfaces. *The Visual Computer* 22, 9 (2006), 865–873. 2
- [LH89] LIU D., HOSCHEK J.:  $G^1$  continuity conditions between adjacent rectangular and triangular bézier surface patches. *Computer-Aided Design* 21, 4 (1989), 194–200. 3
- [Loo87] LOOP C.: Smooth subdivision surfaces based on triangles. *Master's thesis, University of Utah, Department of Mathematics* (1987). 1
- [LP88] LOTT N. J., PULLIN D.: Method for fairing b-spline surfaces. *COMP. AIDED DES.* 20, 10 (1988), 597–604. 2
- [PR97] PETERS J., REIF U.: The simplest subdivision scheme for smoothing polyhedra. *ACM Transactions on Graphics (TOG)* 16, 4 (1997), 420–431. 4
- [Qu90] QU R.: Recursive subdivision algorithms for curve and surface design. *School of Information Systems, Computing and Mathematics* (1990). 1
- [Sze90] SZELISKI R.: Fast surface interpolation using hierarchical basis functions. *Pattern Analysis and Machine Intelligence, IEEE Transactions on* 12, 6 (1990), 513–528. 2
- [YLN96] YE X., LIANG Y., NOWACKI H.: Geometric continuity between adjacent bézier patches and their constructions. *Computer aided Geometric design* 13, 6 (1996), 521–548. 3
- [ZC06] ZHENG J., CAI Y.: Interpolation over arbitrary topology meshes using a two-phase subdivision scheme. *Visualization and Computer Graphics, IEEE Transactions on* 12, 3 (2006), 301–310. 2
- [ZZC01] ZHANG C., ZHANG P., CHENG F. F.: Fairing spline curves and surfaces by minimizing energy. *Computer-Aided Design* 33, 13 (2001), 913–923. 2