Progressive Iterative Interpolation for Subdivision Surfaces

xxx xxx

Abstract. In this paper we give an insight to the interpolation method proposed by [1]. First, we point out that this method is an extension of the progressive iterative approximation of B-spline surface to subdivision surface. We also solve the left open problem in [1]: proving the convergence of the iterative interpolation for subdivision surfaces. Then based on our analysis, we give a more simple and efficient modified interpolation algorithm for subdivision surfaces.

CR Categories: I.3.5 [Computer Graphics]: Computational Geometry and Object Modeling - curve, surface, solid and object representations;

Keywords: subdivision surfaces, Loop subdivision surfaces, Catmull-Clark subdivision surfaces, interpolation

1 Introduction

Subdivision surface is popular now in the fields of Computer Animation, CAD, Geometric Modeling, and so on. The ability to model arbitrary topology surface makes it more suitable than classical spline surfaces in some applications. [3] proposed the Catmull-Clark subdivision surface which is the generalization of bicubic spline surface. [2] designed the Doo-Sabin subdivision method which is the generalization of quadratic spline surface. Later, [9] developed the Loop subdivision for triangle mesh which generalized the Box spline. All these three popular subdivision methods are approximating schemes. The other type of subdivision method is interpolating scheme which interpolates its original mesh. The famous interpolation subdivision method is the butterfly subdivision method proposed in [7] which was modified to generate smoother surfaces in [10]. The interpolating scheme for quadrilateral mesh was proposed in [19].

Interpolation is one of general approaches in surface design and shape modeling. There are plenty of literatures dealing with the interpolation problem with different surface representations. As the appearance of recursive subdivision surface, various interpolation methods based on subdivision surfaces have been developed. One kind of those methods requires solving a global linear system of equations, like [5] and [4]. For the computational cost of solving a large linear system of equations, many researchers developed other methods. [6] proposed an always worked method by using a two-phase subdivision method. The method of [16] avoided the solving a linear equations uses the concept of similarity. And [8] gave one interpolation method avoiding solving a linear system of equations using the quasi-interpolation. Recently, [1] presented a very simple and efficient interpolation approach by just moving the vertices of the mesh. But they didn't provide the proof of its convergence and left it as an open question

In this paper, we give insight to [1]'s method and give the convergence proof of this method. We find that this method could be viewed as the extension of the progressive iteration interpolation for uniform spline proposed in [14] and [11]. This is so called progressive iterative approximation property of uniform B-spline bases. More general results for non-uniform B-spline has been given in [13]. [12] presented that if the given basis is totally positive, and its corresponding collocation matrix is non-singular, this bases have the progressive iteration approximation property. [15] further proved that the normalized B-basis satisfies the progressive iterative approximation property with the fastest convergence rates.

In this paper, we will prove that the subdivision surfaces also satisfy the progressive iterative approximation property. In section 2, we will analyze the convergence for Loop subdivision surface and give a modified algorithm. In section 3, we present the similar result for Catmull-Clark subdivision surface restricted to quadrilateral mesh. Then we will give some examples to investigate the convergence rate and give our conclusion.

2 Iterative interpolation for Loop subdivision surface:

The subdivision surface for triangle mesh was proposed in [9]. It also analyzed its continuity and gave the formula to get the limit point on the limit surface. The limit points of a vertex with valence n as

Figure 1: A vertex with valence n in Loop subdivision

shown in Fig. 1 on the Loop subdivision surface can be calculated through the following formula:

$$V_\infty = \beta_n V + (1 - \beta_n)Q \tag{1}$$

where $\beta_n = \frac{3}{11-8\times\left(\frac{3}{8}+(\frac{3}{8}+\frac{1}{4}\cos\frac{2\pi}{n})^2\right)}$, $Q = \frac{1}{n}\sum_{i=1}^{n} Q_i$.

The essential part of the idea of progressive iterative interpolation is to find a corresponding point on the surface for each interpolated point and use the differences between them to update the control mesh. If the interpolated point is V and the corresponding point on the surface is V^k , the updating process could be written as:

$$e^k = V - V^k$$
$$V^{k+1} = V^k + e^k$$

Based on the above limit point formula, we can construct the iterative fitting algorithm for Loop subdivision surface by using the limit point of each vertex as the corresponding point. Assume the current mesh is M^k , and then we can get a mesh M^{k+1} which is a better approximation to original mesh. For each vertex V^k of M^k, we compute the corresponding limit point $V_\infty^k = \beta_n V^k + (1-\beta_n)Q^k$.Then the difference is calculat as $e^k = V - V^k$. Now every vertex of M^{k+1} is updated by summing the corresponding vertex of M^k and its difference, that is $V^{k+1} = V^k + e^k$. As our below proof shows, the iteration will converge to a mesh which generates a Loop subdivision surface interpolating the original mesh. The details of the iterative interpolation algorithm are presented here:

Iterative Interpolation Algorithm

```
1. Input a triangle mesh $M$ and the maximum error $E_M$;
2. Initialize the $CurrentMaxError = E_M + 1$;
3. $k = 0$;
4. While ($CurrentMaxError > E_M$) {
       For every vertex $V$ in $M^k$ {
           Calculating the corresponding vertex on
           the limit surface $V_\infty^k$;
           $e^k = V - V_\infty^k$;
           $V^{k+1} = V^k + e^k$;
           if ($\|e^k\| > CurrentMaxError$)
               10pt$CurrentMaxError = \|e^k\|$;
       }
5.     $k = k + 1$;
6. }
```

We can prove that e^{k+1} always less than the e^k , so it is not necessary to use the local adjusting in [1]. The method here is much simpler than the algorithm proposed in [1], which needs a complicated shortest point computation. The convergence of our algorithm is analyzed in detail next.

2.1 Convergence of the iterative interpolation for Loop subdivision surface

Before our proof, a fact about the eigenvalues of the product of positive definite matrices needs to be mentioned. We present this fact in Lemma 1.

Lemma 1 *If A and B are positive definite, the eigenvalues of AB are positive.*

The proof of Lemma 1 is quickly followed by the fact that if P and Q are square matrices of order n, then PQ and QP have the same eigenvalues (cf. Magnus and Neudecker [17], 1988, P.14).

We know that the differences of the $k+1$-th step e^{k+1} could be written as:

$$\begin{aligned} e^{k+1} &= V - V_\infty^{k+1} \\ &= V - \big(\beta_n V k + 1 + (1-\beta_n) Q^{k+1}\big) \\ &= V - \big(\beta_n V^{k+1} + (1-\beta_n) Q^{k+1}\big) \\ &\quad - \Big(\beta_n e^k + \frac{1-\beta n}{n} \sum_{i=1}^{n} e_{Q_i}^k\Big) \\ &= e^k - \Big(\beta_n e^k + \frac{1-\beta n}{n} \sum_{i=1}^{n} e_{Q_i}^k\Big) \end{aligned}$$

Thus in the matrix form, we have

$$\big[e_1^{k+1}, e_2^{k+1}, \ldots, e_n^{k+1}\big]^T = (I - B) \begin{bmatrix} e_1^k \\ e_2^k \\ \vdots \\ e_n^k \end{bmatrix}$$

$$\big[e_1^{k+1}, e_2^{k+1}, \ldots, e_n^{k+1}\big]^T = (I - B)^k \begin{bmatrix} e_1^1 \\ e_2^1 \\ \vdots \\ e_n^1 \end{bmatrix}$$

where I is one identity matrix and B is

$$\begin{pmatrix} \beta_{n_1} & \ldots & \frac{1-\beta_{n_1}}{n_1} & \ldots \\ \vdots & \ddots & & \\ \frac{1-\beta_{n_i}}{n_i} & \ldots & \beta_{n_i} & \ldots \\ \vdots & & & \beta_{n_k} \end{pmatrix}$$

The matrix B has the following properties:

(1) $a_{ij} \geq 0$, and $\sum_{j=1}^{n} a_{ij} = 1$, that is $\|B\|_\infty = 1$;

(2) There are $n_i + 1$ positive elements in i-th row, and the positive elements in each row are equal except the element in the diagonal line.

(3) If $a_{ij} = 0$, then $a_{ji} = 0$;

Properties (1) and (2) are simply implied from the formula of e^{k+1} , and property (3) is because the differences of two consecutive levels could be related to each other for their vertices constituting an edge in the mesh. Due to these properties, we can rewrite the matrix B as

$$B = DS$$

where D is diagonal matrix and S is a symmetric matrix

$$\mathbf{D} = \begin{pmatrix} \frac{1-\beta_{n_1}}{n_1} & 0 & \ldots & 0 \\ 0 & \frac{1-\beta_{n_2}}{n_2} & \ldots & 0 \\ \vdots & & \ddots & \\ 0 & & & \frac{1-\beta_{n_k}}{n_k} \end{pmatrix}$$

$$\mathbf{S} = \begin{pmatrix} \frac{n_1\beta_{n_1}}{1-\beta_{n_1}} & \ldots & 1 & \ldots \\ \vdots & \ddots & & \\ 1 & \ldots & \frac{n_i\beta_{n_i}}{1-\beta_{n_i}} & \ldots \\ \vdots & & & \frac{n_k\beta_{n_k}}{1-\beta_{n_k}} \end{pmatrix}$$

D is no problem positive definite for $\frac{1-\beta_{n_i}}{n_i} > 0 (1 \leq i \leq k)$. Now we argue that the matrix S is also positive definite, which plays a key role in our convergence proof.

Proposition 1 *The matrix S is positive definite.*

Proof: To prove S is positive definite, we consider the corresponding quadric form.

$$f(x_1, x_2, \ldots, x_n) = \mathbf{X}^T S \mathbf{X}$$

where $X = (x_1, x_2, \ldots, x_n)^T$. If $f(x_1, x_2, \ldots, x_n)$ is positive for any none zero X, the symmetric matrix S is positive definite.

We notice that if vertices i and j are the endpoints of one edge e_{ij} in the mesh, then $a_{ij} = a_{ji} = 1$ in the matrix S. And an edge is adjacent to two faces exactly in a closed triangle mesh. Hence, we have

$$\begin{aligned} f(x_1, x_2, \ldots, x_n) &= \sum_{\text{all faces}} \frac{1}{2} (x_i + x_j + x_r)^2 \\ &\quad + \sum_{i=1}^{k} \left(\frac{n_i \beta_{n_i}}{1-\beta_{n_i}} - \frac{n_i}{2} \right) x_i^2 \end{aligned}$$

where x_i, x_j, and x_r are the corresponding three vertices of a face in the triangle mesh. From the formula 1, we know $\frac{n\beta_n}{1-\beta_n} \geq \frac{2}{3} n$ for $n \geq 3$. Hence, $f(x_1, x_2, \ldots, x_n)$ is positive for any none zero X. Thus S is positive definite.

Based on the above lemma and proposition, we can easily derive that the iterative interpolation for Loop subdivision is convergent.

Proposition 2 *The iterative fitting algorithm for Loop subdivision surface is convergent.*

Proof: The algorithm is convergent if and only if the absolute value of all eigenvalues of the matrix $P =$

Figure 2: A vertex with valence n in Catmull-Clark subdivision

$I-B$ are less than 1. Hence, if all eigenvalues $\lambda_i(1 \leq i \leq k)$ of B are $0 < \lambda_i \leq 1$, then all eigenvalues of P are $0 \leq 1-\lambda_i < 1(1 \leq i \leq k)$. It means that the algorithm is convergent.

Since $\|B\|_\infty = 1$, we have $\lambda_i \leq 1$. Also we have $B = DS$, where D and S are both positive definite. Hence all eigenvalues of B are positive. Thus, the iterative interpolation algorithm is convergent.

3 Convergence of the iteration interpolation for Catmull-Clark Subdivision surface

Based on the same idea, we can prove the convergence of iterative fitting algorithm for Catmull-Clark Subdivision surface. Firstly, the corresponding limit point of a vertex with valence n in the mesh can be evaluated by the following equality.

$$\begin{aligned} V_\infty &= \frac{n^2}{n(n+5)}V + \frac{4}{n(n+5)}\sum_{i=1}^{n} E_i \\ &\quad + \frac{1}{n(n+5)}\sum_{i=1}^{n} F_i \end{aligned} \tag{2}$$

where E_i is the edge point of vertex V, F_i is are the face points of vertex V. This evaluation formula requires that the faces surrounding vertex V are all quadrilateral face. Because our proof completely depends on this formula, our convergence proof can only be applied on the quadrilateral mesh. It could be viewed as one deficiency of our proof. But in practice the quadrilateral mesh is also very popular, so our method has practical effects as well. Through the formula 2, we can get the relation of the consecutive differences as following:

$$\begin{aligned} e^{k+1} &= V - V_\infty^{k+1} \\ &= V - \left(\frac{n^2}{n(n+5)}V^{k+1} + \frac{4}{n(n+5)}\sum_{i=1}^{n} E_i^{k+1} \right. \\ &\quad \left. + \frac{1}{n(n+5)}\sum_{i=1}^{n} F_i^{k+1}\right) \\ &= V - \left(\frac{n^2}{n(n+5)}V^{k} + \frac{4}{n(n+5)}\sum_{i=1}^{n} E_i^{k} \right. \\ &\quad \left. + \frac{1}{n(n+5)}\sum_{i=1}^{n} F_i^{k}\right) - \left(\frac{n^2}{n(n+5)}e^k \right. \\ &\quad \left. + \frac{4}{n(n+5)}\sum_{i=1}^{n} e_{E_i}^k + \frac{1}{n(n+5)}\sum_{i=1}^{n} e_{F_i}^k\right) \\ &= e^k - \left(\frac{n^2}{n(n+5)}e^k + \frac{4}{n(n+5)}\sum_{i=1}^{n} e_{E_i}^k \right. \\ &\quad \left. + \frac{1}{n(n+5)}\sum_{i=1}^{n} e_{F_i}^k\right) \end{aligned}$$

Then using the matrix form, we have

$$\left[e_1^{k+1}, e_2^{k+1}, \ldots, e_n^{k+1}\right]^T = (I-C)\begin{bmatrix} e_1^k \\ e_2^k \\ \vdots \\ e_n^k \end{bmatrix}$$

$$\left[e_1^{k+1}, e_2^{k+1}, \ldots, e_n^{k+1}\right]^T = (I-C)^k\begin{bmatrix} e_1^1 \\ e_2^1 \\ \vdots \\ e_n^1 \end{bmatrix}$$

Model	# of vertices	# of iterations	Max Error	Ave Error
Horse	233	12	0.000912016	0.00022811
Bird	1129	9	0.000766811	8.8345e-5
Hand	6191	3	0.000914111	0.000164207
Boy	17342	6	0.000913795	9.5615e-5

Table 1: Interpolating results for Loop subdivision surface.

where I is identity matrix and C is

$$\begin{pmatrix} \frac{n_1^2}{n_1(n_1+5)} & \cdots & \frac{4}{n_1(n_1+5)} & \cdots & \frac{1}{n_1(n_1+5)} & \cdots \\ \vdots & \ddots & & & & \\ \frac{4}{n_i(n_i+5)} & & \frac{n_i^2}{n_i(n_i+5)} & & & \\ \vdots & & & \ddots & & \\ \frac{1}{n_j(n_j+5)} & & & & \frac{n_j^2}{n_j(n_j+5)} & \\ \vdots & & & & & \ddots \end{pmatrix}$$

The matrix C could write as the product of two symmetric matrices.

$$C = HM$$

where H and M are

$$\mathbf{H} = \begin{pmatrix} \frac{1}{n_1(n_1+5)} & 0 & \cdots & 0 \\ 0 & \frac{1}{n_2(n_2+5)} & \cdots & 0 \\ \vdots & & \ddots & \\ 0 & & & \frac{1}{n_k(n_k+5)} \end{pmatrix}$$

$$\mathbf{M} = \begin{pmatrix} n_1^2 & \cdots & 4 & \cdots & 1 & \cdots \\ \vdots & \ddots & & & & \\ 4 & & n_i^2 & & & \\ \vdots & & & \ddots & & \\ 1 & & & & n_j^2 & \\ \vdots & & & & & \ddots \end{pmatrix}$$

For the matrix M, we know that if vertex v_i is the face point of v_j, then v_j is also one face point of v_i. This shows that if $a_{ij} = 1$, then $a_{ji} = 1$. Similarly, if $a_{ij} = 4$, then $a_{ji} = 4$ because one edge shares two endpoints. Obviously, we have $\|C\|_\infty = 1$. Thus the eigenvalues of C are not bigger than one. If we can prove that matrix M is positive definite, it is easy to show the iterative fitting algorithm for Catmull-Clark subdivision surface is convergent by following the same way for Loop subdivision surface. Actually, the matrix M is indeed positive definite. We give our proof here.

Proposition 3 *The matrix M is positive definite.*

Proof: The above observation shows why the matrix M is symmetric. But these observations also give the hints for our proof. Using the definition of positive definite matrix, we write down the quadric form for M.

$$f(x_1, x_2, \ldots, x_k) = X^T M X$$

where $X = (x_1, x_2, \ldots, x_k)^T$. Thus we can rewrite $f(x_1, x_2, \ldots, x_k)$ as

$$\begin{aligned} f &= \sum_{\text{all faces}} (x_i + x_j + x_k + x_r) \\ &\quad + \sum_{\text{all edges}} (x_i + x_j) + \sum_{\text{all vertices}} \left(n_i^2 - 3n_i\right) x_i^2 \end{aligned}$$

We also know $n_i \geq 3$, hence $n_i^2 - 3n_i \geq 0$. This means $f(x_1, x_2, \ldots, x_k) \geq 0$ for any none zero X. Thus, M is **AT LEAST semipositive definite**.

Following the same way of proving the convergence for Loop subdivision surface, we can easy know that the iterative interpolation algorithm for Catmull-Clark subdivision surface is also convergent. We conclude this in the following proposition.

Proposition 4 *The iterative fitting algorithm for Catmull-Clark subdivision surface is convergent.**(NOT COMPLETELY CORRECT !)***

4 Results

In this section, we give several examples. Table 1 shows the iteration numbers, maximum and average errors when using Loop subdivision surfaces. As pointed out in [1], progressive iterative interpolation method is very efficient and can handle very huge meshes, and its complexity is only $O(mn)$ where n is the number of vertices of the original mesh. Although no fairness control factor is added in progressive iterative interpolation, the results show that it could produce visually pleasing surface easily. So far the progressive iterative interpolation for Catmull-Clark subdivision surface requires that the mesh is quadrilateral mesh. Except this limitation, the progressive iterative interpolation works very well. Figure 7 is the result of this method applied on a mushroom mesh with 226 vertices. The maximum error for the mushroom mesh is 0.000813821, and the average error is 0.000235506.

5 Conclusion and future work

We have proved the convergence of iterative interpolation method for Loop and Catmull-Clark subdivision surfaces. This method could be treated as an extension of the progressive iterative approximation property from spline to subdivision. Based on our analysis, we give a modified progressive iterative interpolation algorithm. This modified algorithm is simpler and more efficient. But our proof to Catmull-Clark subdivision surface is only valid on the quadrilateral mesh. To extend our proof to arbitrary mesh for Catmull-Clark subdivision surface is our next concern. How to make this method effective for open mesh is also an interesting topic.

Acknowledgement

All the triangle meshes in this paper are from the Princeton Shape Benchmark [18] and the quadrilateral mesh mushroom is downloaded from internet. We appreciate those.

References

[1] T. Maekawa, Y. Matsumoto, K. Namiki. Interpolation by Geometric Algorithm. Computer-Aided Design 2007; 39 (4):313-323.

[2] Doo D, Sabin M. Behaviour of recursive division surfaces near extraordinary points. Computer-Aided Design 1978; 10(6):356-60.

[3] Catmull E, Clark J. Recursively generated B-spline surfaces on arbitrary topological meshes. Computer-Aided Design 1978; 10(6):350-5

[4] Halstead M, Kass M, DeRose T. Efficient, fair interpolation using CatmullCClark surfaces. In: Proceedings of SIGGRAPH 1993. 1993. p. 47-61

[5] Nasri, AH. Surface interpolation on irregular networks with normal conditions. Computer Aided Geometric Design 1991;8:89-96

[6] Zheng J, Cai YY. Interpolation over arbitrary topology meshes using a two-phase subdivision scheme. IEEE Transactions on Visualization and Computer Graphics 2006;12(3):301-10

[7] N. Dyn, D. Levin, and J.A. Gregory. A Butterfly Subdivision Scheme for Surface Interpolation with Tension Control. ACM Trans. Graphics, vol. 9, no. 2, pp. 160-169, Apr. 1990.

[8] N. Litke, A. Levin, and P. Schröder. Fitting Subdivision Surfaces. Pro Visualization 2001, pp. 319-324.

[9] C. Loop. Smooth Subdivision Surfaces Based on Triangles. Master'thesis, Dept. of Math., Univ. of Utah, 1987

[10] D. Zorin, P. Schroder, and W. Sweldens. Interpolating Subdivision for Meshes with Arbitrary Topology. Computer Graphics, Ann. Conf. Series, vol. 30, pp. 189-192 1996.

[11] de Boor, C. 1979. How does Agee's method work? In: Proceedings of the 1979 Army Numerical Analysis and Computers Conference, ARO Report 79-3, Army Research Office, pp. 299-302

[12] Lin, H., Bao, H., Wang, G. Totally positive bases and progressive iteration approximation. Computer & Mathematics with Applications, 2005; 50:575-58

[13] Lin, H.,Wang, G., Dong, C. Constructing iterative non-uniform B-spline curve and surface to fit data points. Science in China (Series E), 2003;33:912-923 (in Chinese)

[14] Qi, D., Tian, Z., Zhang, Y., Zheng, J.B. The method of numeric polish in curve fitting. Acta Mathematica Sinica 1975; 18: 173-184 (in Chinese)

[15] J. Delgado , J.M. Peña. Progressive iterative approximation and bases with the fastest convergence rates. Computer Aided Geometric Design 2007; 24 (1):10-18

[16] S. Lai, F. Cheng. Similarity based Interpolation using Catmull-Clark Subdivision Surfaces. The Visual Computer 2006; 22 (9):865-873.

[17] I.R. Magnus, H. Neudecker. Matrix Differential Calculus with Applications in Statistics and Econometrics. New York: John Wiley & Sons.

[18] P. Shilane, P. Min, M. Kazhdan, and T. Funkhouser. The Princeton Shape Benchmark. Shape Modeling Int'l, 2004

[19] L. Kobbelt. Interpolatory Subdivision on Open Quadrilateral Nets with Arbitrary Topology. Comput. Graph. Forum 1996;15(3): 409-420.

(a) The original mesh with 233 vertices

(b) Interpolating Loop surface

Figure 3: Loop subdivision surface interpolating the horse mesh after 12 iterations.

(a) The original mesh with 1129 vertices

(b) Interpolating Loop surface

Figure 4: Loop subdivision surface interpolating the bird mesh after 9 iterations.

(a) The original mesh with 6191 vertices

(b) Interpolating Loop surface

Figure 5: Loop subdivision surface interpolating the hand mesh after 3 iterations.

(a) The original mesh with 17342 vertices

(b) Interpolating Loop surface

Figure 6: Loop subdivision surface interpolating the boy mesh after 6 iterations.

(a) The original mesh with 226 vertices

(b) Interpolating Loop surface

Figure 7: Catmull-Clark subdivision surface interpolating the mushroom mesh after 8 iterations.