Subdivision based Interpolation with Shape Control

Fengtao Fan
University of Kentucky
Deparment of Computer Science
Lexington, KY 40506, USA
ffan2@uky.edu

Fuhua (Frank) Cheng
University of Kentucky
Deparment of Computer Science
Lexington, KY 40506, USA
cheng@cs.uky.edu

Shuhua Lai
Virginia State University
Department of Mathematics and Computer Science
Petersburg, VA 23806
slai@vsu.edu

Abstract

An interpolation technique with the capability of local shape control for meshes of arbitrary topology is presented. The interpolation is a progressive process which iteratively updates the given mesh, through a two-phase Doo-Sabin subdivision scheme, until a control mesh whose limit surface interpolates the given mesh is reached. For each iteration of the progression, the two-phase scheme works by first applying a modified Doo-Sabin subdivision to the input mesh and then applying the regular Doo-Sabin subdivision to the resulting mesh. The modified Doo-Sabin subdivision carries a parameter for each face of the input mesh. These parameters provide required freedom to adjust the interpolating subdivision surface at the user's command. Local shape control is possible. It is proved that the progressive interpolation process converges for any parameters between 0 *and* 1*. Therefore, this is a well-defined process. The progressive interpolation process satisfies both the local and global properties. Hence, the new technique can handle meshes of any size and is very faithful and efficient. Test cases that show the effectiveness of the new technique are included.*

1. Introduction

Subdivision surfaces are becoming popular in areas such as computer animation, geometric modeling, and computer games, because of their capability in representing any complex shape with only one surface. A subdivision surface is generated by repeatedly refining (subdividing) a given control mesh until a limit surface is reached. Therefore, a subdivision surface is determined by the subdivision scheme. Subdivision schemes in general fall into two categories: *approximating schemes* and *interpolating schemes*. An approximating scheme generates a smooth limit surface that approximates the original mesh. Catmull-Clark subdivision scheme [1], Doo-Sabin subdivision scheme [4] and Loop subdivision scheme [13] are typical approximating schemes. An interpolating scheme generates a smooth limit surface that interpolates the original mesh. The famous butterfly subdivision method [5] and its modified version [21], and Kobbelt's subdivision scheme for quadrilateral meshes [8] are interpolating subdivision schemes.

Interpolation is a frequently used technique in shape modeling and design. The idea is to construct a surface to interpolate the vertices of a given mesh (sometimes also the derivatives or normals at the vertices) so that the shape of the surface, guided by the mesh, would be close to the shape that we want to design or model. Traditional interpolation techniques can not handle design or modeling of objects with complicated topology because of restriction imposed by the rectangular domains of the surface representations. One usually needs to decompose the objects into several components and perform interpolation on these components separately. Subdivision surfaces do not have such a restriction. Therefore, interpolation using subdivision surfaces is a more promising design and modeling technique.

An intuitive approach to interpolate using subdivision surfaces is simply performing an interpolating subdivision scheme such as [21] or [8] on the given mesh. In this approach, new vertices are defined as local affine combinations of nearby vertices only. Therefore, this approach possesses properties of a local method and, consequently, can handle meshes with large number of vertices. However, since no vertex is ever moved once it is computed, any distortion in the early stage of the subdivision will persist. This makes interpolating subdivision very sensitive to irregularity in the given mesh. In addition, it is difficult for this approach to interpolate normals or derivatives.

A less intuitive approach is to use approximating subdivision schemes in the construction of the interpolating surfaces. This approach is a global method because it needs to solve a global system of linear equations to find control mesh of the interpolating surface [16][7]. Therefore, it can reproduce the shape of the data mesh faithfully, but can not handle meshes with large number of vertices. To avoid solving a large system of linear equations, several alternatives have been proposed such as quasi-interpolation [12], similarity based interpolation [9] and two-phase subdivision scheme [19]. However, a subdivision based interpolation method that has the advantages of both a local method and a global method is not available yet.

In this paper, we present a new subdivision based interpolation technique with the capability of local shape control for meshes of arbitrary topology. The construction of the interpolating subdivsion surface is through a progressive process which iteratively upgrading the vertices of the given mesh, through a two-phase Doo-Sabin subdivision scheme, until a control mesh whose limit surface interpolates the given mesh is reached. For each iteration of the progression, the two-phase scheme works by first applying a modified Doo-Sabin subdivision to the input mesh and then applying the regular Doo-Sabin subdivision to the resulting mesh. The modified Doo-Sabin subdivision carries a parameter for each face of the input mesh. These parameters provide required freedom to adjust the interpolating subdivision surface at the user's command. Local shape control is possible. It is proved that the progressive interpolation process converges for any parameters between 0 and 1. Therefore, this is a well-defined process. The limit of the progressive process has a global form while the vertex upgrading process is a local operation. Therefore the progressive interpolation process enjoys advantages of both a local method and a global method. Test cases show that the new technique is indeed faithful and efficient and can handle meshes of any size.

The remaining part of the paper is arranged as follows. The new two-phase Doo-Sabin subdivision scheme is introduced in Section 2. A progressive interpolation process whose updating step is built on top of this scheme is introduced in Section 3. Convergence of the progressive interpolation process is proved in Section 4. Implementation issues and test results are discussed and presented in section 5. Concluding remarks are given in Section 6.

2. Two-phase Doo-Sabin Subdivision Scheme

Given a control mesh M^0 with arbitrary topology, a Doo-Sabin subdivision surface is generated by iteratively refining the control mesh until a smooth limit surface is reached [4]. The limit surface is called a subdivision surface because the mesh refining process is a generalization of the quadratic B-spline surface subdivision scheme. Therefore, Doo-Sabin subdivision surfaces include quadratic B-spline surfaces as special cases.

If M^i is the resulting mesh after the i-th refinement step, the $i+1$-st refinement step is performed as follows. For each face $\mathbf{F} = \mathbf{V}_1\mathbf{V}_2\ldots\mathbf{V}_f$ in M^i, a new vertex $\mathbf{V}_i^{'}$ is generated for each old vertex of the face $\mathbf{V}_i$ through the following formula:

$$\mathbf{V}_i^{'} = \sum_{j=1}^{f} \alpha_{ij}\mathbf{V}_j \qquad i = 1, 2, \ldots, f;$$

where α_{ij} are defined as follows:

$$\alpha_{ij} = \begin{cases} \frac{f+5}{4f} & i = j \\ \frac{3+2\cos\left(2(i-j)\pi/f\right)}{4f} & i \neq j \end{cases}$$

The new vertices are then connected to form faces of the new mesh M^{i+1} using the following rules.

1. New vertices generated for each face are connected to form an *F-face*;

2. New vertices generated along an old edge of the mesh M^i are connected to form an *E-face*;

3. New vertices generated around an old vertex of M^i are connected to form a *V-face*.

The valence of each new vertex is four. But side numbers of new faces are usually different (except that E-faces which are always quadrilaterals). However, once a face is created, all the F-faces subsequently created within that face will always have the same number of sides. An important property of the new faces is, the centroid of each new face lies on the limit surface. This property is frequently used in the construction of an interpolating Doo-Sabin subdivision surface.

Typical subdivision based interpolation techniques do not provide the user with the option of shape control. To add freedom for shape control, we propose a two-step scheme for Doo-Sabin subdivision. A two-step scheme for Catmull-Clark subdivision was first used in [3] to design an always working interpolation method, where a single parameter λ was introduced in the first subdivision step. The two-step Doo-Sabin subdivision scheme to be introduced here is more general and has the ability of local shape control. The new subdivision scheme carries a shape parameter for each face in its first subdivision step. These parameters provide the required freedom in shape control at the user's command. Assume the faces of the mesh M^0 are ordered from 1 to $|F|$. For each face $\mathbf{F} = \mathbf{V}_1\mathbf{V}_2 \ldots \mathbf{V}_f$ of M^i, a new vertex $\mathbf{V}_i^{'}$ is also generated for each old vertex of the face $\mathbf{V}_i$. However, the generation of the new vertices depends on a parameter assigned to that face as follows.

$$\mathbf{V}_i^{'} = \lambda \mathbf{V}_i + (1-\lambda)\mathbf{A} \tag{1}$$

where λ is a parameter between 0 and 1, and $\mathbf{A}$ is the centroid of the face: $\mathbf{A} = \frac{1}{f}(\mathbf{V}_1 + \mathbf{V}_2 + \ldots + \mathbf{V}_f)$. F-Faces, E-face, and V-faces of M^1 are then created following the same rules as those specified above. Once the new mesh M^1 is created, phase-one subdivision is done. We then perform regular Doo-Sabin subdivision on the new mesh M^1 iteratively to generate a limit surface. The surfaces in figure 1 are generated using the new two-phase scheme. The limit surface would resemble the initial mesh closely if shape parameters attached to the faces of the initial mesh are close to 1.

3. Progressive interpolation

Given a rectangular mesh M^0 to be interpolated. By viewing M^0 as the control mesh of a B-spline surface, one can compute the distances between vertices of M^0 and corresponding points on the B-spline surface. If these distances are added to the vertices of M^0, one gets a new mesh M^1 whose B-spline surface is closer to the vertices of M^0. By computing distances between vertices of M^0 and the corresponding points on the B-spline surface of M^1, and adding these distances to the vertices of M^1, one gets a new mesh M^2 whose B-spline surface is even closer to the vertices of M^0. Iteratively repeat this process, one gets a sequence of meshes M^i whose corresponding B-spline surfaces converge to a limit surface that interpolates M^0. This is the basic idea of *progressive interpolation* originally proposed for B-splines [10, 17]. An attempt to use this technique to interpolate meshes with arbitrary topology using Loop subdivision surfaces was recently made in [14]. However, it couldn't prove convergence of the corresponding progressive process. In the following, we present a progressive interpolation technique for Doo-Sabin subdivision surfaces. The vertex upgrading process is driven by the two-phase subdivision scheme defined in the previous section. Therefore, a user can control the shape of the interpolating surface by adjusting values of the shape parameters carried by phase-one subdivision. Proof of convergence of the progressive process is given in the next section.

Give an initial mesh M^0 with arbitrary topology, by performing a phase-one Doo-Sabin subdivision on M^0, one gets a new mesh M^1. For each vertex $\mathbf{V}$ of M^0, there is a corresponding *V-face* in M^1. The centroid of the V-face lies on the limit surface of M^1. Hence, centroids of the V-faces can serve as limit points of M^0's vertices. Figure 2 shows the neighborhood of $\mathbf{V}$. $\mathbf{V}$ has n adjacent edges and faces. The adjacent edge points are denoted $\mathbf{E}_1$, $\mathbf{E}_2$, …, $\mathbf{E}_n$. Vertices in the i-th face are denoted $\mathbf{F}_1^i$, $\mathbf{F}_2^i$, …, $\mathbf{F}_{f_i-3}^i$ where f_i is the number of vertices in the i-th face. $\mathbf{W}_i^{'}$ denotes the new vertex in the i-th face after one subdivision that is corresponding to $\mathbf{V}$. Then $\mathbf{W}_1^{'}$, $\mathbf{W}_2^{'}, \ldots,$ $\mathbf{W}_n^{'}$ are the vertices of the V-face corresponding to the vertex $\mathbf{V}$. The limit point of $\mathbf{V}$ can be computed as follows.

$$\mathbf{V}^\infty = \frac{1}{n}\sum_{i=1}^{n}\mathbf{W}_i^{'}$$

From eq. (1), we can expand V^∞ as follows.

$$\begin{aligned} \mathbf{V}^\infty &= \frac{1}{n}\sum_{i=1}^{n}\mathbf{W}_i^{'} \\ &= \frac{1}{n}\sum_{i=1}^{n}(\lambda_i + \frac{1-\lambda_i}{f_i})\mathbf{V} \\ &\quad + \frac{1}{n}\sum_{i=1}^{n}(\frac{1-\lambda_i}{f_i} + \frac{1-\lambda_{i-1}}{f_{i-1}})\mathbf{E}_i \\ &\quad + \frac{1}{n}\sum_{i=1}^{n}\sum_{j=1}^{f_i-3}\frac{1-\lambda_i}{f_i}\mathbf{F}_j^i \end{aligned} \tag{2}$$

In general, let M^k be the resulting mesh after the k-th iteration, and $\mathbf{S}^k$ the limit surface of the two-phase Doo-Sabin subdivision process. For each vertex $\mathbf{V}^k$ of M^k, first compute the corresponding limit point $\mathbf{V}^\infty$ on $\mathbf{S}^k$ by equation (2). Then compute $\mathbf{D}^k$, the difference between the vertex $\mathbf{V}^0$ in the initial mesh and $\mathbf{V}^\infty$.

$$\mathbf{D}^k = \mathbf{V}^0 - \mathbf{V}^\infty \tag{3}$$

This distance is then added to the vertex $\mathbf{V}^k$ to get the new vertex $\mathbf{V}^{k+1}$ for M^{k+1}.

$$\mathbf{V}^{k+1} = \mathbf{V}^k + \mathbf{D}^k$$

As the iteration proceeds, the updated mesh converges to a mesh whose limit surface interpolates the initial mesh. The computation of limit points is the key operation in each iteration. This computation process, according to eq. (2), is direct and is a local operation. Thus the progressive interpolation process has the advantages of a local method, like the ability to handle large meshes. On the other hand, since the limit of the modified meshes is a global system, the progressive interpolation process also has the advantages of a global method. Hence the new progressive interpolation

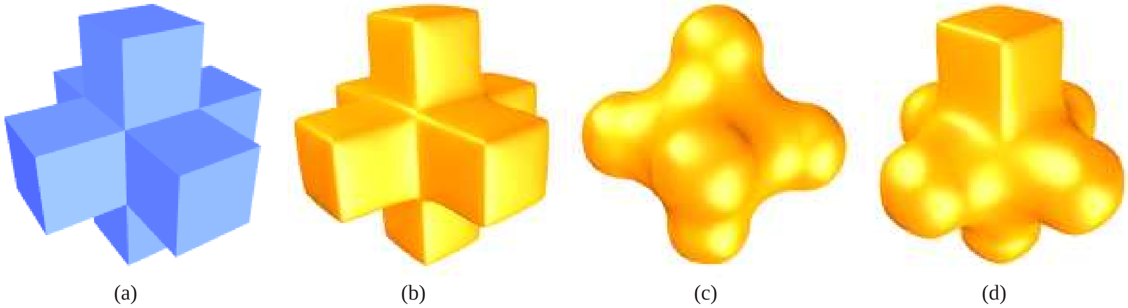


Figure 1. Limit surfaces with different shape parameters for two-phase Doo-Sabin subdivision scheme: (a) the initial mesh; (b) limit surface with all shape parameters set to 0.8; (c) limit surface with all shape parameters set to 0.4; (d) limit surface with the shape parameters of the upper part set to 0.8 and others to 0.4.

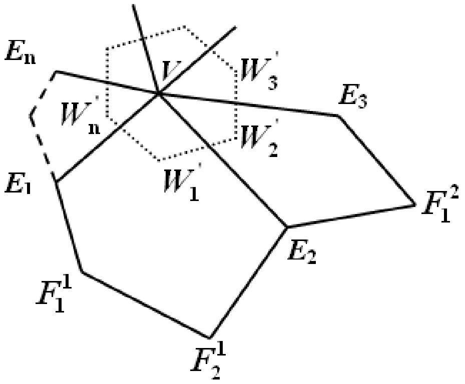


Figure 2. Neighborhood of vertex V

process has the advantages of both a local method and a global method. We next show that the new progressive interpolation process converges as long as the shape parameters are between 0 and 1. Hence, this is a well-defined process.

4. Convergence of the Progressive Interpolation Method

From equation (2) and equation (3), we have the following relationship between $\mathbf{D}^k$ and $\mathbf{D}^{k-1}$.

$$\begin{aligned}
\mathbf{D}^k &= \mathbf{V}^0 - \mathbf{V}^\infty \\
&= \mathbf{V}^0 - \frac{1}{n}\sum_{i=1}^{n} \Big((\lambda_i + \frac{(1-\lambda_i)}{f_i})\mathbf{V}^k \\
&\qquad + \frac{(1-\lambda_i)}{f_i}(\mathbf{E}_i^k + \mathbf{E}_{i+1}^k) \\
&\qquad + \frac{(1-\lambda_i)}{f_i}\sum_{j=1}^{f_i-3}(\mathbf{F}_j^i)^k \Big) \\
&= \mathbf{D}^{k-1} - \frac{1}{n}\sum_{i=1}^{n} \Big(\\
&\qquad (\lambda_i + \frac{(1-\lambda_i)}{f_i})\mathbf{D}^{k-1} \\
&\qquad + \frac{(1-\lambda_i)}{f_i}(\mathbf{D}_{\mathbf{E}_i}^{k-1} + \mathbf{D}_{\mathbf{E}_{i+1}}^{k-1}) \\
&\qquad + \frac{(1-\lambda_i)}{f_i}\sum_{j=1}^{f_i-3}\mathbf{D}_{\mathbf{F}_j^i}^{k-1}\Big)
\end{aligned} \tag{4}$$

Eq. (4) can be put in a compact matrix form as follows.

$$\begin{aligned}
\left[\mathbf{D}_1^k, \mathbf{D}_2^k, \ldots, \mathbf{D}_m^k\right]^T &= (I - B)\begin{bmatrix} \mathbf{D}_1^{k-1} \\ \mathbf{D}_2^{k-1} \\ \vdots \\ \mathbf{D}_m^{k-1} \end{bmatrix} \\
&= (I - B)^k \begin{bmatrix} \mathbf{D}_1^{0} \\ \mathbf{D}_2^{0} \\ \vdots \\ \mathbf{D}_m^{0} \end{bmatrix}
\end{aligned}$$

where m is the number of vertices in the given mesh, I is an identity matrix and B is an $m \times m$ matrix

$$B = \left[\frac{B_{ij}}{n_i}\right]_{i,j=1}^{m}$$

with

$$B_{i,j} = \begin{cases}
\sum_{k=1}^{n_i}(\lambda_k + \frac{(1-\lambda_k)}{f_k}), \quad i = j \\[2ex]
(\frac{1-\lambda_k}{f_k} + \frac{1-\lambda_{k-1}}{f_{k-1}}), \quad \mathbf{V}_j \; is \; the \; k^{th} \; Edge \\
\qquad\qquad\qquad\qquad point \; of \; \mathbf{V}_i \\[2ex]
\frac{1-\lambda_k}{f_k}, \quad \mathbf{V}_j \; is \; a \; vertex \; of \; the \; k^{th} \\
\qquad\qquad Face \; of \; \mathbf{V}_i \\[2ex]
0, \quad otherwise
\end{cases} \tag{5}$$

Each row of matrix B is computed from the eq. (2). B is the matrix used for computing the limit points on the limit surface. B can be decomposed as the product of a diagonal matrix Σ and a symmetric matrix S as follows

$$B = \Sigma S$$

where

$$\Sigma = \begin{bmatrix} \frac{1}{n_1} & 0 & \ldots & 0 \\ 0 & \frac{1}{n_2} & \ldots & 0 \\ \vdots & & \ddots & \\ 0 & & & \frac{1}{n_m} \end{bmatrix}$$

and

$$S = [B_{ij}]_{i,j=1}^{m}$$

with $B_{i,}$ defined in (5). Note that the relationship between two edge vertices or two face vertices is symmetric. Hence, it is easy to see that S is symmetric. Actually, S is positive definite.

Proposition 1 *The matrix S is positive definite if $\lambda_i \in (0,1),\ i = 1, 2, \ldots, |F|$.*

Proof: Consider the quadratic form of the matrix S:

$$f(X) = X^T S X$$

If $f(X) > 0$ for any nonzero X, then the symmetric matrix S is positive definite. From eq. (2), face point in the j-th face surrounding $\mathbf{V}$ has coefficient $\frac{1-\lambda_j}{f_j}$ and edge point of $\mathbf{V}$ has coefficient $\frac{1-\lambda_j}{f_j} + \frac{1-\lambda_{j-1}}{f_{j-1}}$ when the edge is shared by the j-th face and the $(j-1)$-st face. Note that these coefficients depend on the face only. Then we have

$$\begin{aligned}
f(X) = &\sum_{\text{all faces}} \frac{1-\lambda_j}{f_j}(\mathbf{V}_1 + \mathbf{V}_2 + \ldots + \mathbf{V}_{f_j})^2 \\
&+ \sum_{\text{all vertices}} (\sum_{i=1}^{n} \lambda_i)\mathbf{V}^2
\end{aligned}$$

Since the parameter λ_j for each face is between 0 and 1, $\frac{1-\lambda_j}{f_j} > 0$ and $\sum_{i=1}^{n} \lambda_i > 0$ always hold. Thus $f(X) > 0$

holds for any $X \neq 0$. Hence, the matrix S is always positive definite if shape parameter for each face is in $(0, 1)$.

The diagonal matrix Σ is obviously symmetric positive definite. Then the fact that the eigenvalues of B are positive follows from the following lemma.

Lemma 1 *Eigenvalues of the product of positive definite matrices are positive.*

The proof of Lemma 1 follows immediately from the fact that if P and Q are square matrices of the same dimension, then PQ and QP have the same eigenvalues (see, e.g., [15], p.14).

Convergence of the progressive process can be proven easily now.

Proposition 2 *The progressive interpolation process driven by the two-phase Doo-Sabin subdivision scheme is convergent if* $\lambda_i \in (0, 1)(i = 1, 2, \ldots, |F|)$.

Proof: First, it is clear that $\|B\|_\infty = 1$. Thus every eigenvalue μ of B satisfies $|\mu| \leq 1$. We also know that the eigenvalues of B are positive. Therefore, the eigenvalues of B satisfy the condition $0 < \mu \leq 1$. But then the eigenvalues of $I - B$, $1 - \mu$, satisfy $0 \leq 1 - \mu < 1$. Hence the progressive interpolation process is convergent.

5. Results

The parameters carried by phase-one subdivision provide the freedoms to control the shape of the interpolating surface. These parameters act as a tension parameter when their values are close to 1. Therefore the shape of the interpolating surface resembles that of the given mesh when the shape parameters are close to 1. Figure 3(b) shows an example with shape parameter set to 0.8 for each face. The resulting interpolating surface is visually very pleasing when the shape parameters are around 0.5. Figure 3(c) shows an example with shape parameter set to 0.4 for each face. With two-phase subdivision, local shape control is also possible. Figure 3(d) shows an example where shape parameters of the four faces defining the north angle are set as 0.8 with all the other shape parameters set to 0.5. At the same time, the progressive interpolation process is very efficient and can handle large meshes easily because only local affine operations are required in each iteration. Our test cases show that the progressive interpolation process indeed converges quickly.

Several examples are presented in the following figures, showing both the given mesh and the resulting interpolating surfaces in each case. We use relative error, instead of absolute error, to define the threshold in stopping the iteration. The threshold is set to 0.01% of the bounding box diagonal of the initial mesh in all the test cases. Table 1 gives the comprehensive data of these examples.

6. Concluding Remarks

A novel progressive interpolation process driven by a two-phase Doo-Sabin subdivision scheme is presented. Phase-one subdivision of the two-phase scheme carries a shape parameter for each face of the given mesh. Therefore, in addition to having the advantages of both a local method and a global method, this technique also allows a user to control the shape of the interpolating surface interactively. Actually, since the shape parameters are independently defined, shape control can be done both locally and globally. The progressive interpolation process converges for shape parameters between 0 and 1. So we finally have a well-defined subdivision driven progressive interpolation process with the capability of both global and local shape control. Currently, the presented technique is designed for closed meshes only. A future research direction is to consider this technique for open meshes.

7. Acknowledgement

Research work presented in this paper is supported by NSF (DMI-0422126) and KSTC (COMM-Fund-712). Triangular meshes used in this paper are downloaded from the Princeton Shape Benchmark [18].

References

[1] Catmull E, Clark J, Recursively generated B-spline surfaces on arbitrary topological meshes, *Computer-Aided Design* 1978, 10(6):350-355.

[2] de Boor C, How does Agee's method work? *Proc. 1979 Army Numerical Analysis and Computers Conference*, ARO Report 79-3, Army Research Office, 299-302.

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

[4] Doo D, Sabin M, Behaviour of recursive division surfaces near extraordinary points, *Computer-Aided Design* 1978, 10(6):356-360.

[5] Dyn N, Levin D, Gregory JA, A Butterfly Subdivision Scheme for Surface Interpolation with Tension Control, *ACM Trans. Graphics* 1990, 9(2):160-169.

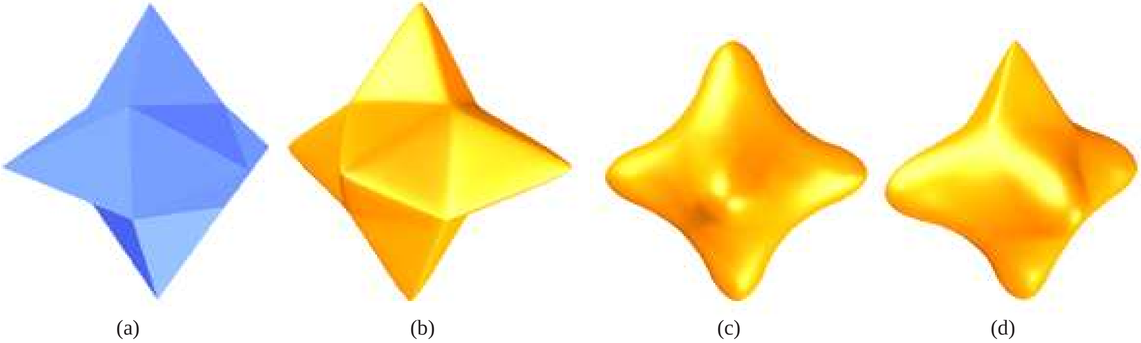


Figure 3. Interpolating surfaces with different shape paramters using two-phase Doo-Sabin subdivision scheme: (a) given mesh; (b) interpolating surface with all shape parameters set to 0.8; (c) interpolating surface with all shape parameters set to 0.4; (d) interpolating surface with local shape control: shape parameters of the upper part set to 0.8 and others to 0.4.

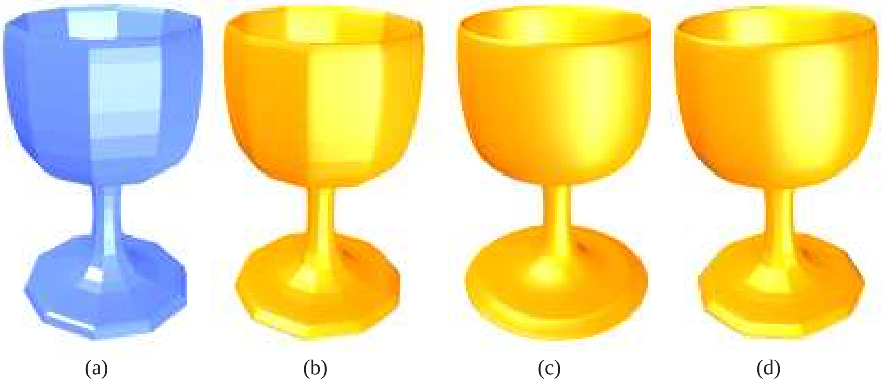


Figure 4. Interpolating surfaces with different shape paramters using two-phase scheme for Doo-Sabin subdivision: (a) given mesh; (b) interpolating surface with all shape parameters set to 0.85; (c) interpolating surface with all shape parameters set to 0.4; (d) interpolating surface with local shape control: shape parameters of the bottom part set to 0.85 and others to 0.4.

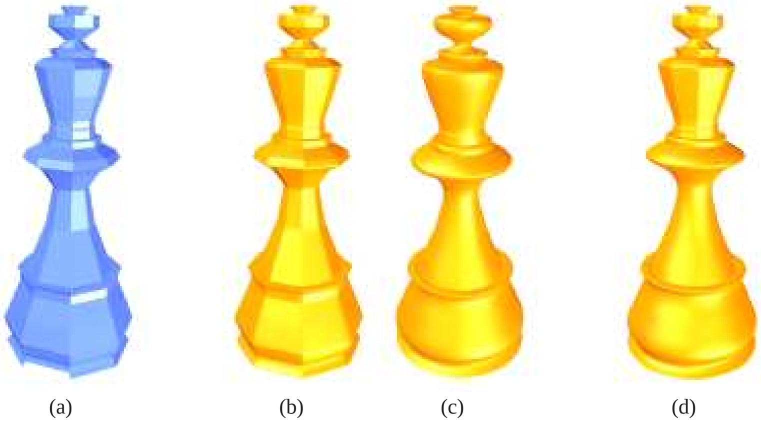


Figure 5. Interpolating surfaces with different shape paramters using two-phase Doo-Sabin subdivision scheme: (a) given mesh; (b) interpolating surface with all shape parameters set to 0.85; (c) interpolating surface with all shape parameters set to 0.5; (d) interpolating surface with local shape control: shape parameters of the upper part set to 0.85 and others to 0.5.

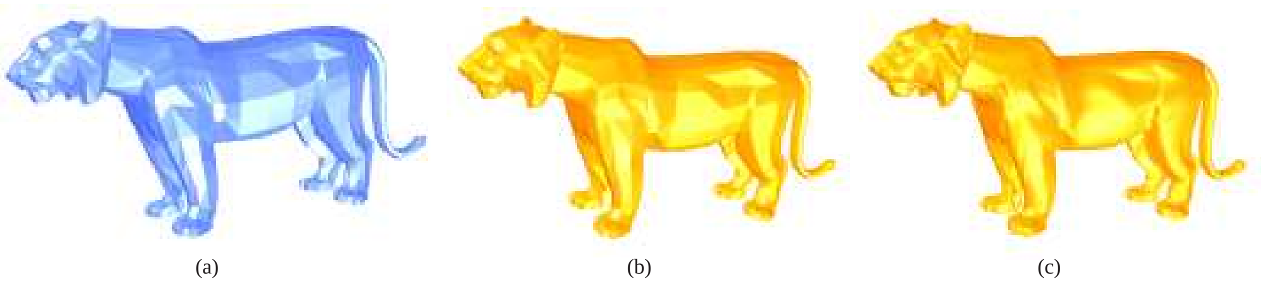
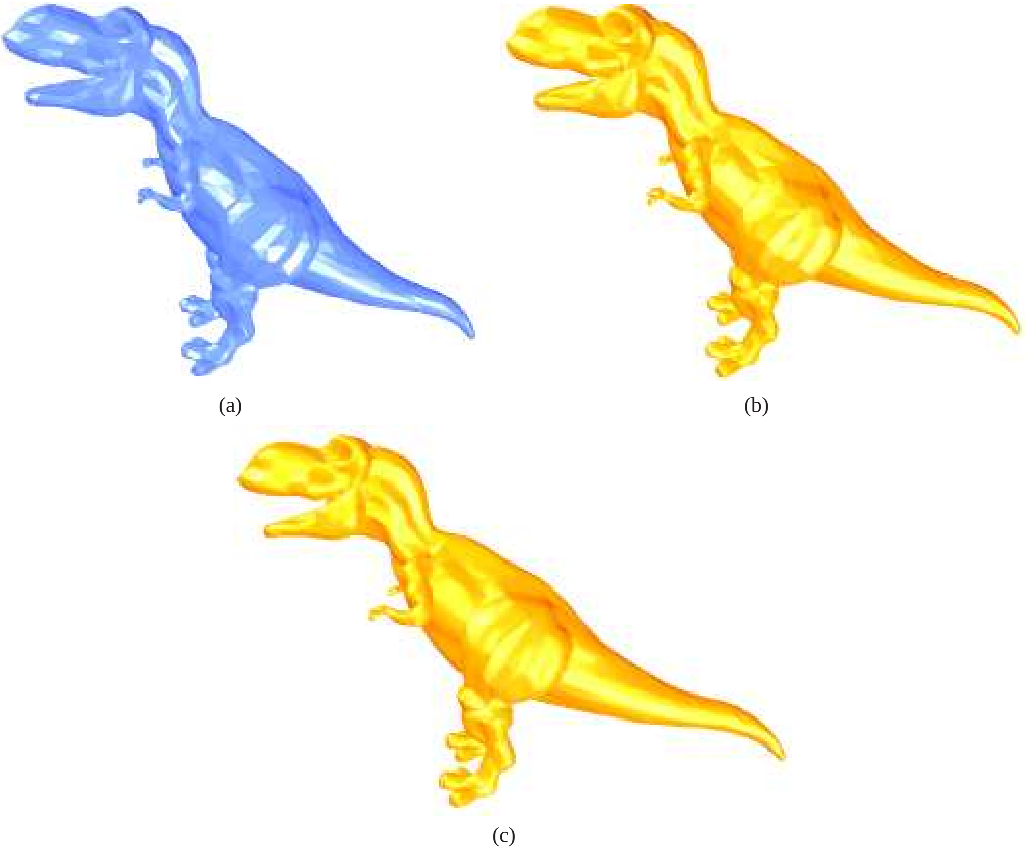


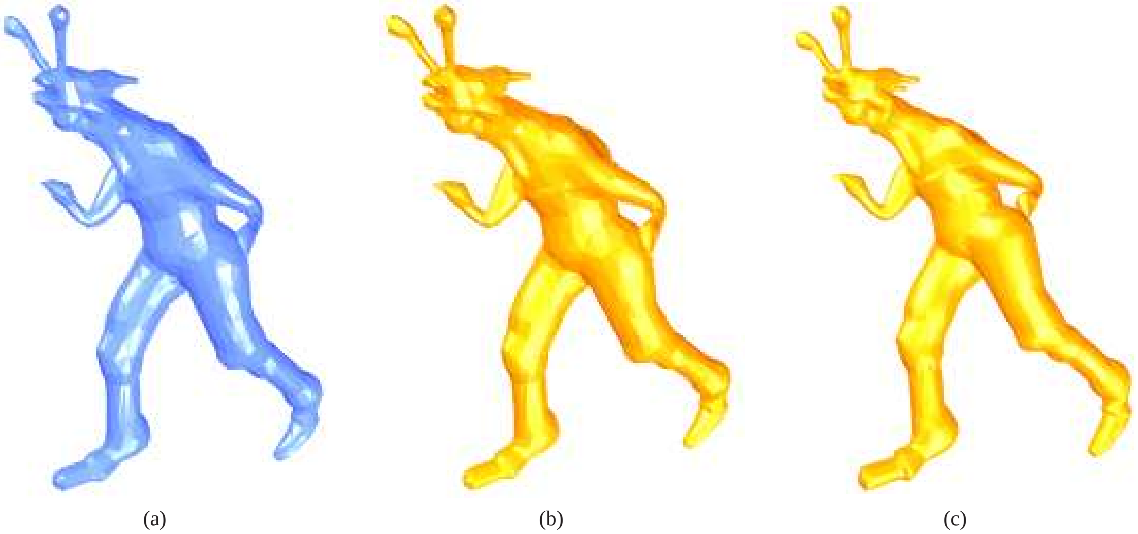
Figure 6. (a) given mesh; (b) interpolating surface with all shape parameters set to 0.85; (c) interpolating surface with all shape parameters set to 0.5.



(a) (b)

(c)

Figure 7. (a) given mesh; (b) interpolating surface with all shape parameters set to 0.85; (c) interpolating surface with all shape parameters set to 0.5.



(a) (b) (c)

Figure 8. (a) given mesh; (b) interpolating surface with all shape parameters set to 0.85; (c) interpolating surface with all shape parameters set to 0.5.

Table 1. Progressive iterative interpolation: test results.

Figures	# of vertices	# of iterations	Max Error	Ave Error
4(b)	520	4	3.48051e-006	2.39273e-007
4(c)	520	15	1.06938e-005	1.04466e-006
4(d)	520	8	8.51077e-006	9.14787e-007
5(b)	314	4	0.0147411	0.0047881
5(c)	314	10	0.0160123	0.00624637
5(d)	314	10	0.0160123	0.00386015
6(b)	956	4	5.43858e-006	1.42327e-006
6(c)	956	10	6.23132e-006	1.60614e-006
7(b)	1590	4	5.69258e-006	1.02823e-006
7(c)	1590	10	6.37943e-006	1.20608e-006
8(b)	1200	4	6.00313e-006	1.46748e-006
8(c)	1200	10	6.85102e-006	1.68769e-006

[6] Gerald Farin, *Curves and Surfaces for CAGD A Practical Guide, 5th edition*, Morgan-Kaufmann, 2001.

[7] Halstead M, Kass M, DeRose T, Efficient, fair interpolation using CatmullCClark surfaces, *Proc.SIGGRAPH 1993*, 47-61.

[8] Kobbelt L, Interpolatory Subdivision on Open Quadrilateral Nets with Arbitrary Topology, *Comput. Graph. Forum* 1996, 5(3):409-420.

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

[10] Lin H, Wang, G, Dong C, Constructing iterative non-uniform B-spline curve and surface to fit data points, *Science in China (Series F)*, 2004, 47:315-331.

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

[12] Litke N, Levin A, Schröder P, Fitting Subdivision Surfaces, *Proc. Visualization* 2001, 319-324.

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

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

[15] Magnus IR, Neudecker H, *Matrix Differential Calculus with Applications in Statistics and Econometrics*, New York, John Wiley & Sons, 1988.

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

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

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

[19] Zheng J, Cai Y, Making Doo-Sabin surface interpolation always work over irregular meshes, *The Visual Computer* 2005, 21(4):242-251.

[20] Zheng J, Cai YY, Interpolation over arbitrary topology meshes using a two-phase subdivision scheme, *IEEE Trans. Visualization and Computer Graphics* 2006, 12(3):301-310.

[21] Zorin D, Schroder P, Sweldens W, Interpolating Subdivision for Meshes with Arbitrary Topology, *Computer Graphics, Ann. Conf. Series*, 1996, 30:189-192.