Parametrization of General Catmull-Clark Subdivision Surfaces and its Applications

Shuhua Lai[1] and Fuhua (Frank) Cheng[2]

[1]University of Kentucky, shuhua@csr.uky.edu
[2]University of Kentucky, cheng@cs.uky.edu

ABSTRACT

A new parametrization technique and its applications for general Catmull-Clark subdivision surfaces are presented. The new technique extends J. Stam's work by redefining all the eigen basis functions in the parametric representation for general Catmull-Clark subdivision surfaces and giving each of them an explicit form. The entire eigenstructure of the subdivision matrix and its inverse are computed exactly and explicitly with no need to precompute anything. Therefore, the new representation can be used not only for evaluation purpose, but for analysis purpose as well. The new approach is based on an Ω-*partition* of the parameter space and a detoured subdivision path. This results in a block diagonal matrix with constant size diagonal blocks (7×7) for the corresponding subdivision process. Consequently, eigen decomposition of the matrix is always possible and is simpler and more efficient. Furthermore, since the number of eigen basis functions required in the new approach is only one half of the previous approach, the new parametrization is also more efficient for evaluation purpose. This is demonstrated by several applications of the new techniques.

Keywords: subdivision, Catmull-Clark surfaces, parametrization.

1. INTRODUCTION

Subdivision surfaces have become popular recently in graphical modeling and animation because of their capability in modeling/representing complex shape of arbitrary topology [6], their relatively high visual quality, and their stability and efficiency in numerical computation. Subdivision surfaces can model/represent complex shape of arbitrary topology because there is no limit on the shape and topology of the control mesh of a subdivision surface. With the parametrization technique for subdivision surfaces becoming available [14] and with the fact that non-uniform B-spline and NURBS surfaces are special cases of subdivision surfaces becoming known [12], we now know that subdivision surfaces cover both *parametric forms* and *discrete forms*. Parametric forms are good for design and representation, discrete forms are good for machining and tessellation. Hence, we have a representation scheme that is good for all graphics and CAD/CAM applications.

Research work for subdivision surfaces has been done in several important areas, such as surface interpolation [8, 19-22], surface evaluation [4, 13-16], surface trimming [9], boolean operations [3], and mesh editing [17]. However, powerful evaluation and analysis techniques for subdivision surfaces have not been fully developed yet. Parametrization methods that have been developed so far are suitable for evaluation purpose only, not for analysis purpose, because these methods either do not have an explicit expression, or are too complicated for each part to be explicit. For instance, in [14], eigen functions are pre-computed numerically and stored in a file. So they can be used for evaluation purpose only. Note that exact evaluation at a point of a subdivision surface is possible only if there is an explicit parametrization of the surface. Hence, an explicit parametrization is not only critical for analysis purpose, but for evaluation and rendering purpose as well.

In this paper we will present an Ω-*partition* based approach to solve several important problems of subdivision surfaces: (1) computation of new control vertices at a specified subdivision level, (2) explicit parametrization of an extra-ordinary patch, and (3) surface evaluation at arbitrary parameter space point with eigen functions computed on the fly. The new approach is based on the observation that the subdivision process on the control vertices can be broken into subdivision processes on smaller, same frequency groups after a few linear transformations. Using a different ordering of the vertices and the idea of enlarging the subdivision matrix, the subdivision matrix can be transformed into a block matrix with each block being circulant [1, 18]. Hence it is natural to use the Fourier matrices

to transform them into diagonal matrices. Each such subdivision process on points of the same frequency is independent of the *valence* of the extra-ordinary vertex. The dimension of the corresponding subdivision matrix for each frequency group is 7×7. Therefore, the process of using a large subdivision matrix to perform the subdivision process on the control vertices can be replaced with a set of 7×7 matrices on the same frequency groups. This not only makes computation of the eigenstructures of the subdivision matrices always possible, but also simpler and more efficient. Inverses of the eigenvector matrices can also be explicitly computed.

2. PREVIOUS WORK

2.1 Catmull-Clark Subdivision Surfaces

Given a control mesh, a *Catmull-Clark subdivision surface* (CCSS) is generated by iteratively refining the control mesh [5]. The limit surface is called a *subdivision surface* because the mesh refining process is a generalization of the uniform B-spline surface *subdivision technique*. The *valence* of a mesh vertex is the number of mesh edges adjacent to the vertex. A mesh vertex is called an *extra-ordinary vertex* if its valence is different from four. Vertex V in Fig. 1(a) is an extra-ordinary vertex of valence five. A mesh face with an extra-ordinary vertex is called an *extra-ordinary face*. The *valance* of an extra-ordinary face is the valence of its extra-ordinary vertex.

Given an extra-ordinary face S = $S_{0,0}$. If the valence of its extra-ordinary vertex is *n*, then the surface patch corresponding to this extra-ordinary face is influenced by $2n+8$ control vertices. The control vertices shown in Fig. 1(a) are the ones that influence the patch marked with an "S = $S_{m-1,0}$". In general, if $S_{m-1,0}$ is the extra-ordinary subpatch generated after *m-1* subdivision steps, then by performing a Catmull-Clark subdivision step on the control vertices of $S_{m-1,0}$, one gets $2n+17$ new control vertices. See Fig. 1(b) for the new control vertices generated for the patch $S_{m-1,0}$ shown in (a). These $2n+17$ new control vertices define four subpatches: $S_{m,b}$, *b=0, 1, 2, 3* (See Fig. 1(b)). $S_{m,0}$ is again an extra-ordinary patch but $S_{m,1}$, $S_{m,2}$, and $S_{m,3}$ are regular uniform bicubic B-spline patches. Iteratively repeat this process, one gets a sequence of regular bicubic B-spline patches $S_{m,b}$, *m ≥ 1, b=1,2,3*, a sequence of extra-ordinary patches $S_{m,0}$, *m ≥ 0*, and a sequence of extra-ordinary vertices. The extra-ordinary patches converge to the limit point of the extra-ordinary vertices [8]. The regular bicubic B-spline patches $S_{m,b}$, *m ≥ 1, b=1,2,3*, and the limit point of the extra-ordinary vertices form a partition of S.

2.2 Previous Parametrization and Evaluation Methods

An algorithm for the evaluation of a subdivision surface at an arbitrary point was first proposed by J. Stam in 1998 for Catmull-Clark subdivision surfaces [14] and then in 1999 for Loop subdivision surfaces [15]. Stam's approach shows that an extra-ordinary surface patch and its derivatives can be represented as a linear combination of the control points with weights defined by a set of $2n+8$ eigenbasis functions where *n* is the valence of the extra-ordinary patch. The representation satisfies simple scaling relations and can be easily evaluated in constant time. However, even though analytical expressions for the eigenbasis functions have been derived, some of them are too complicated to be reported in the paper [14]. Besides, some of the eigenbasis functions are redundant. We will show in this paper that only *n+6* eigenbasis functions are actually needed and, consequently, the evaluation process can be made more efficient. J. Stam's approach is mainly developed for evaluation purpose. As we shall present, our parametrization results can be used not only for evaluation, but for analysis purpose as well.

Warrent and Weimer presented a method in [18] for computing all eigenvalues and eigenvectors of the subdivision matrix by writing the subdivision matrix for the 2-ring in block circulant form. Ball and Storry [1] also used the similar approach to compute the eigenstructure of the subdivision matrix. However, as far as we know, the inverse of the matrix of the eigenvectors has never been computed explicitly, and the overall explicit eigenstructure has never been integrated into the parameterization formula. In this paper, based on the eigenanalysis results of [1], an explicit and exact evaluation formula is derived.

Zorin extended the work of J. Stam by considering subdivision rules for piecewise smooth surfaces with parameter-controlled boundaries [16]. The main contribution of their work is the usage of a different set of basis vectors for the evaluation process which, unlike eigenvectors, depend continuously on the coefficients of the subdivision rules. The advantage of this algorithm is that it is possible to define evaluation for parametric families of rules without considering excessive number of special cases, while improving numerical stability of calculation. In addition to Stam's approach, two different parametrizations of Catmull-Clark subdivision surfaces have been proposed by Boier-Martin and Zorin [4]. The motivation of their work is to provide parametrization techniques that are differentiable everywhere. Although all the natural parameterizations of subdivision surfaces are not *C*-1around extraordinary vertices of valence higher than four [4], the resulting surfaces are still *C-2* almost everywhere. Moreover, despite of the fact that the partial derivatives diverge around an extraordinary vertex, in this paper, we will show that

an standardized normal vector can be calculated explicitly *everywhere*. As we know, precisely calculated normal vector is indispensable for surface shading purposes.

Exact evaluation of piecewise smooth Catmull-Clark surfaces near sharp and semi-sharp features is considered in [13]. Constant-time performance is achieved by employing Jordan decomposition of the subdivision matrix. In this paper we will show that special features can be generated using ordinary Catmull-Clark rules with constant-time evaluation performance as well.
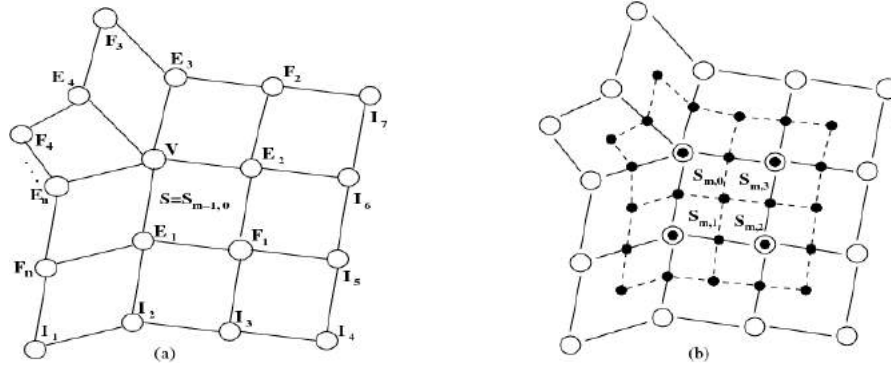

Fig. 1. (a) Control vertices that influence an extra-ordinary patch. (b) New control vertices (solid dots) generated after a Catmull-Clark subdivision.

3. PARAMETRIZATION OF A PATCH

The regular bicubic B-spline patches $S_{m,b}$, $m \geq 1$, $b = 1,2,3$, induce a partition on the unit square $[0,1] \times [0,1]$. The partition is defined by: $\{\Omega_{m,b}\}$, $m \geq 1$, $b=1,2,3$, with

$$\Omega_{m,1} = [\frac{1}{2^m}, \frac{1}{2^{m-1}}] \times [0, \frac{1}{2^m}], \qquad \Omega_{m,2} = [\frac{1}{2^m}, \frac{1}{2^{m-1}}] \times [\frac{1}{2^m}, \frac{1}{2^{m-1}}], \qquad \Omega_{m,3} = [0, \frac{1}{2^m}] \times [\frac{1}{2^m}, \frac{1}{2^{m-1}}]$$

(see Fig. 2(a) for an illustration of the partition [14]). For any (u, v) in $[0,1] \times [0,1]$ but $(u, v) \neq (0, 0)$, there is an $\Omega_{m,b}$ that contains (u, v). To find the value of S at (u, v), first map $\Omega_{m,b}$ to the unit square. If (u,v) is mapped to $(\overline{u}, \overline{v})$ by this mapping, then compute the value of $S_{m,b}$ at $(\overline{u}, \overline{v})$. The value of S at $(0,0)$ is the limit of the extra-ordinary vertices. For convenience of subsequent reference, the above partition will be called an Ω-*partition* of the unit square. In the above process, m and b can be computed as follows:

$$m(u,v) = \min\left\{ \left\lceil \log_{\frac{1}{2}} u \right\rceil, \left\lceil \log_{\frac{1}{2}} v \right\rceil \right\}, \qquad b(u,v) = \begin{cases} 1, & if \quad 2^m u \geq 1 \quad and \quad 2^m u < 1 \\ 2, & if \quad 2^m u \geq 1 \quad and \quad 2^m u \geq 1 \\ 3, & if \quad 2^m u < 1 \quad and \quad 2^m u \geq 1 \end{cases}$$

The mapping from $\Omega_{m,b}$ to the unit square is defined as: $(u,v) \rightarrow (\overline{u}, \overline{v}) = (\phi(u), \phi(v))$, where

$$\phi(t) = \begin{cases} 2^m t, & if \quad 2^m t \leq 1 \\ 2^m t - 1, & if \quad 2^m t > 1 \end{cases} \tag{1}$$

Since each $S_{m,b}$ is a standard B-spline surface, it can be expressed as $S(u,v) = W^T(\overline{u}, \overline{v})MG_{m,b}$, where $G_{m,b}$ is the control point vector of $S_{m,b}$, $W(u, v)$ is a vector containing the *16* power basis functions:

$$W^T(u,v) = [1, u, v, u^2, uv, v^2, u^3, u^2 v, uv^2, v^3, u^3 v, u^2 v^2, uv^3, u^3 v^2, u^2 v^3, u^3 v^3],$$

and M is the B-spline coefficient matrix. An important observation is, $W^T(\overline{u}, \overline{v})$ can be expressed as the product of $W^T(u, v)$ and two matrices: $W^T(\overline{u}, \overline{v}) = W^T(u, v)K^m D_b$, where K is a diagonal matrix

$$K = Diag(1,2,2,4,4,4,8,8,8,8,16,16,16,32,32,64),$$

and D_b is an upper triangular matrix depending on b only. D_b can be obtained by replacing $\overline{u}, \overline{v}$ in $W^T(\overline{u}, \overline{v})$ with $\emptyset(u)$, $\emptyset(v)$ defined in Eq. (1). Therefore, we have $S(u, v) = W^T(u, v)K^m D_b MG_{m,b}$. The computation of the control vertices of $S_{m,b}$ involves two matrices, A and $\overline{A}$ [14]. $\overline{A}$ is a $(2n+17) \times (2n+8)$ matrix, representing the subdivision process shown in Fig. 1(b). A is a $(2n+8) \times (2n+8)$ submatrix of $\overline{A}$, representing the process of mapping the *2n+8*

control vertices of the given extra-ordinary patch to the *2n+8* control vertices of its extra-ordinary subpatch. Let $G =$ [V, E$_1$, …, E$_n$, F$_1$, …, F$_n$, I$_1$, …,I$_7$], then G (See Fig. 1(a) for its labeling) is the column vector representing the control vertices of S. By applying A to G (m-1) times we get the *2n+8* control vertices of the extra-ordinary subpatch S$_{m-1,0}$. Now by applying $\overline{A}$ to the control vertices of Sm-1,0 (represented as G_{m-1}), we get *2n+17* new control points which include the *2n+8* control vertices of S$_{m,0}$. Let $\overline{G}_m$ be the column vector representation of these *2n+17* vertices, we have $\overline{G}_m = \overline{A}\, G_{m-1} = \overline{A}A^{m-1}G$. Then by multiplying $\overline{G}_m$ with an appropriate "picking" matrix P$_b$, we get the control vertices of the subpatch S$_{m,b}$: $G_{m,b} = P_b\overline{G}_m = P_b\overline{A}A^{m-1}G$. Hence we have

$$S(u,v) = W^T(u,v)K^m D_b M P_b \overline{A} A^{m-1}G. \tag{2}$$

This is a parametrization of an extra-ordinary patch. However, this is a costly process to use because it involves *m-1* multiplications of the *(2n+8)×(2n+8)* matrix A. In the next section, we will present an efficient approach to calculate $G_{m,b}$ for any *b* and *m*.

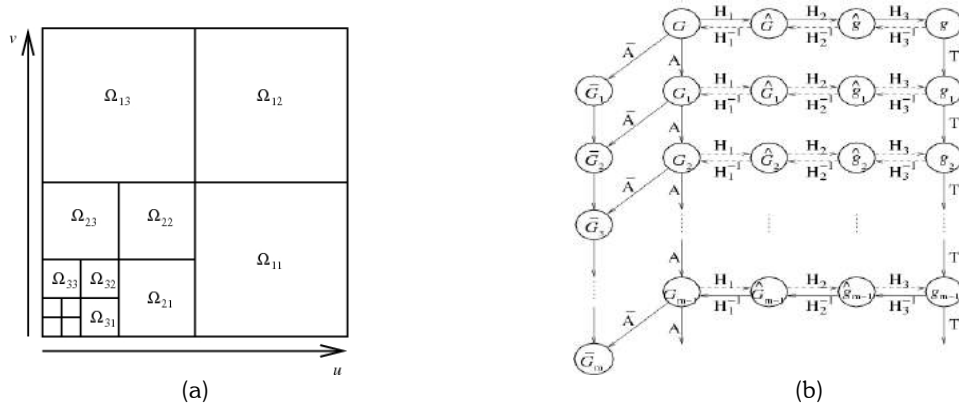
Fig. 2. (a) Ω-partition of the unit square [14]; (b) The extended subdivision diagram.

4. CALCULATE CONTROL VERTICES AFTER *m* SUBDIVISION

The goal here is to show that instead of using the direct path from G to G_{m-1} to compute $G_{m-1}=A^{m-1}G$ in the above equation, one should use the indirect, longer path ($G \rightarrow g \rightarrow g_{m-1} \rightarrow G_{m-1}$) in Fig. 2(b) to do the job. The reason for doing so is: the corresponding matrix *T* is a block diagonal matrix with each diagonal block of dimension 7×7 only. Therefore, the process of computing their eigen decompositions is not only always possible, but also much simpler and more efficient. Details of this new approach and definitions of related mappings are given below. We consider a general CCSS here. That is, the new *vertex point* V' after one subdivision is computed as follows:

$$V' = \alpha_n V + \beta_n \sum_{i=1}^n E_i + \gamma_n \sum_{i=1}^n F_i \tag{3}$$

where α_n, β_n and γ_n are positive numbers and their sum equals one. New *face points* and *edge points* are computed the same way as in [5]. First, to prepare G for the major transformation, we extend G into a vector of seven equal-length components, called $\hat{G}$: $\hat{G} = (V^T, E^T, F^T, I_1^T, I_2^T, I_3^T, I_4^T)^T$, where

$$V = (V, V, \cdots, V)^T, \qquad E = (E_1, E_2, \cdots E_n)^T, \qquad F = (F_1, F_2, \cdots, F_n)^T,$$
$$I_4 = (I_4, 0, 0, \cdots, 0)^T, \quad I_k = (I_k, I_{k+4}, 0, \cdots, 0)^T, \quad k = 1,2,3$$

with all of them having the same length of *n*. We can get $\hat{G}$ from G by a simple extension matrix H$_1$, i.e., $\hat{G} = H_1 G$. Note that the matrix inducing $\hat{G}_i$ to $\hat{G}_{i+1}$, i.e., H$_1$AH$_1$, is a 7n×7n block matrix with each block ($n\times n$) being circulant [1, 18]. Therefore, each of these blocks can be diagonalized exactly using the discrete Fourier transform. Let $\hat{g}$ be the result of applying the discrete Fourier transform L to the components of $\hat{G}$:

$$\hat{g} = (LV^T, LE^T, LF^T, LI_1^T, LI_2^T, LI_3^T, LI_4^T)^T = (v^T, e^T, f^T, i_1^T, i_2^T, i_3^T, i_4^T)^T$$

Each component of $\hat{g}$ has the same length n, but is indexed from 0 to n-1. We can get $\hat{g}$ from $\hat{G}$ by combining all L's into a single matrix H_2, i.e., $\hat{g} = H_2\hat{G}$. It is easy to see that H_2 is a block diagonal matrix. If we re-arrange the elements of $\hat{g}$ into a set of same frequency groups: $g = (h_0^T, h_1^T, \cdots, h_{n-1}^T)^T$, where $h_\omega = (v_\omega, e_\omega, f_\omega, i_{1\omega}, i_{2\omega}, i_{3\omega}, i_{4\omega})^T$, with $0 \le \omega \le n-1$. We can get g from $\hat{g}$ through a $7n$ x $7n$ permutation matrix H_3, i.e., $g=H_3\hat{g}$. The above relationships hold for g_j, G_j, $\hat{g}_j$ and $\hat{G}_j$, $j \ge 1$, as well (See Fig. 2(b)). Since H_1, H_2 and H_3 are invertible, we can easily calculate g_j and G_j from each other.

For each $j \ge 1$, the subdivision process performed on G_{j-1} to get G_j can be reflected on g_{j-1} and g_j through H_1, H_2 and H_3. The induced subdivision process [1] on g_{j-1} can be represented by a $7n \times 7n$ matrix T as: $g_j = Tg_{j-1} = T^j g$. T is a block diagonal matrix with each diagonal block T_ω ($\omega = 0,1,2,...,n-1$), being a $7n \times 7n$ matrix. The expression of each T_ω can be found in [1]. Therefore, for each $m \ge 1$, we have (See Fig. 2(b)): $A^{m-1} = H_1^{-1}H_2^{-1}H_3^{-1}T^{m-1}H_3H_2H_1$.

By combining the above expression with Eq. (2), we have

$$S(u,v) = W^T K^m D_b MP_b \overline{A}H_1^{-1}H_2^{-1}H_3^{-1}T^{m-1}H_3H_2H_1G \tag{4}$$

For a given (u,v), every matrix in Eq. (4) is known to us if valance n is known. Hence it can be used to exactly and explicitly evaluate the position of $S(u,v)$.

5. EIGENANALYSIS OF T

Equation (4) provides a formal parametrization of an extra-ordinary patch. This parametrization, however, is still costly to evaluate because it involves m-1 multiplications of the matrix T. The evaluation process can be considerably simplified if T is decomposed as $T=X^{-1}\Lambda X$, where Λ is a diagonal matrix of eigenvalues of T and X is an invertible matrix whose columns are the corresponding eigenvectors. Therefore, the evaluation of T^{m-1} becomes the evaluation of $X^{-1}\Lambda^{m-1}X$ only. Note that T is a block diagonal matrix. To find the eigen decomposition of T, we first find the eigen decomposition of each diagonal block T_ω of T: $T_\omega = X_\omega^{-1}\Lambda_\omega X_\omega$, $(\omega = 0,1,\cdots,n-1)$. Since each diagonal block T_ω is of size $n \times n$, its eigen decomposition can be calculated explicitly. X, Λ and X^{-1} are then formed as block diagonal matrices with diagonal blocks being X_ω, Λ_ω and X_ω^{-1}, respectively. Consequently, $S(u,v)$ can be expressed as:

$$S(u,v) = W^T K^m Z_b \Lambda^{m-1} ZG \tag{5}$$

where $Z=XH_3H_2H_1$ and $Z_b = D_b MP_b \overline{A} Z^{-1}$. For any given n, these matrices are known explicitly.

There are totally $n+6$ different eigenvalues in Λ. These different eigenvalues of T are:

$$\lambda_0 = 4\alpha_n - 1 + \sqrt{16\alpha_n^2 - 8\alpha_n + 8\beta_n - 3})/8 \qquad \lambda_1 = 4\alpha_n - 1 - \sqrt{16\alpha_n^2 - 8\alpha_n + 8\beta_n - 3})/8$$

$$\lambda_{2\omega} = (c_\omega + 5 + \sqrt{c_\omega^2 + 10c_\omega + 9})/16 \qquad \lambda_{2\omega+1} = (c_\omega + 5 - \sqrt{c_\omega^2 + 10c_\omega + 9})/16$$

$$\lambda_{n+1} = 1, \quad \lambda_{n+2} = 1/8, \quad \lambda_{n+3} = 1/16, \qquad \lambda_{n+4} = 1/32, \quad \lambda_{n+5} = 1/64.$$

where $1 \le \omega \le n/2$, $c_\omega = \cos(2\pi\omega/n)$ and α_n and β_n are defined in Eq. (3). It is easy to check that $\lambda_0 > \lambda_1$ and $\lambda_2 > \lambda_i$ for $3 \le i \le n$.

6. EVALUATION OF A CCSS PATCH

Now we show how Eq. (5) can be used in the efficient evaluation of a CCSS patch at a given (u,v). Eq. (5) can be used for both extra-ordinary and regular patches because the derivation of Eq. (5) did not use the assumption that $n \ne 4$. First note that $S(u, v)$ defined in Eq. (5) can be written as a linear combination of these different eigenvalues in Λ to the $(m$-$1)$st power: $S(u,v) = W^T K^m \sum \lambda_j^{m-1}(Z_b \Theta_j Z)G$, where Θ_j is a $7n \times 7n$ matrix with all the entries being zero except the ones corresponding to λ_j in matrix Λ. Those entries of Θ_j are 1. Let $M_{b,j} = Z_b \Theta_j Z$, We get

$$S(u,v) = W^T K^m \sum_{j=0}^{n+5} \lambda_j^{m-1} M_{b,j}G. \tag{6}$$

The exact expressions of $M_{b,j}$ are shown in the complete version of the paper [7]. Eq. (6) is the most important result of this paper. This equation can be used to evaluate a CCSS patch at any point (including (0,0)), and it can also be used to compute the derivative of a CCSS patch at any point (including (0,0) as well). The patch can be regular or extra-ordinary. Note that for any $m \geq 0$, we have $W^T(u, v)K^m = W^T(2^m u, 2^m v)$. Define

$$\Phi_{b,j}(u,v) = W^T(2^m u, 2^m v)\lambda_j^{m-1}M_{b,j}; \quad \Phi_b(u,v) = \sum_{j=0}^{n+5}\Phi_{b,j}(u,v).$$

$\emptyset_{b,j}(u, v)$ are called the jth *eigen basis function* of CCSSs. There are totally $n+6$ eigen basis functions and for any given (u, v), every eigen basis function can be exactly and explicitly represented. It is easy to check that all the eigen basis functions satisfy the so called *scaling relation* [14, 16]: $\Phi_{b,j}(u/2, v/2) = \lambda_j \Phi_{b,j}(u,v)$. With the above definition, Eq. (6) can be represented as $S(u, v) = \emptyset_b(u, v)G$, which is used for fast rendering in our implementation. In addition, one can compute the derivatives of $S(u, v)$ to any order simply by differentiating $W(u, v)$ in Eq. (6) accordingly. For example,

$$\frac{\partial}{\partial u}S(u,v) = (\frac{\partial W}{\partial u})^T K^m \sum_{j=0}^{n+5}\lambda_j^{m-1}M_{b,j}G \qquad (7)$$

7. EXPLICIT ANALYSIS

7.1 Limit Point

Eq. (6) not only can be used for evaluation purpose, but analytic derivation as well. For example, one gets the limit point of an extra-ordinary vertex simply by setting $u=v=0$ and $m\rightarrow\infty$ in Eq. (6), which generalizes Eq. (13) of [8]:

$$S(0,0) = [1,0,\cdots,0] \cdot M_{b,n+1} \cdot G.$$

7.2 Partial Derivatives

It is known the first partial derivatives of $S(u, v)$ at $(0,0)$ diverge in a natural parametrization [4]. However, knowing the directions of them is sufficient in many applications. As pointed out by [1], when $\lambda_0 \geq \lambda_2$, a general Catmull-Clark subdivision surface is not C-1 continuous. Suppose $\lambda_0 < \lambda_2$, dividing both sides of Eq. (7) by $2^m \lambda_2^{m-1}$, and by setting $u=v=0$ and $m\rightarrow\infty$, we get

$$D_u(0,0) = [0,1,0,0,\cdots,0] \cdot M_{2,2} \cdot G, \qquad D_v(0,0) = [0,0,1,0,\cdots,0] \cdot M_{2,2} \cdot G,$$

where D_u and D_v are the direction vectors of $\frac{\partial S(0,0)}{\partial u}$ and $\frac{\partial S(0,0)}{\partial v}$, respectively. The normal vector at $(0,0)$ is the cross product of them. Similarly, when $\lambda_0 < \lambda_2$, it is easy to calculate the second partial derivatives at $(0,0)$. These derivatives are listed as follows.

$$D_{uu}(0,0) = [0,0,0,2,0,0,0,\cdots,0] \cdot M_{2,2} \cdot G; \quad D_{vv}(0,0) = [0,0,0,0,0,2,0\cdots,0] \cdot M_{2,2} \cdot G;$$

$$D_{uv}(0,0) = [0,0,0,0,1,0,0,\cdots,0] \cdot M_{2,2} \cdot G;$$

where D_{uu}, D_{uv} and D_{vv} are the direction vectors of $\frac{\partial^2 S(0,0)}{\partial u^2}$, $\frac{\partial^2 S(0,0)}{\partial u \partial v}$ and $\frac{\partial^2 S(0,0)}{\partial v^2}$, respectively. Since $M_{2,2}$ is explicitly and exactly known, all these vectors can be calculated once G is given.

7.3 Proof of Tangent Plane Continuity

With the explicit expressions of partial derivatives of $S(u, v)$ at $(0,0)$, some properties of CCSS at an extra-ordinary point can be proved easily. For instance, one can prove that when $\lambda_0 < \lambda_2$, there exists a common tangent plane at an extra-ordinary point.

The tangent plane continuity property has been proven by many people with different approaches [1, 8, 10, 11]. Here a simple proof using our parametrization results is given below.

Expand D_u and D_v, we have $D_u = \sum_{i=1}^{n}\bar{e}_i \cdot E_i + \sum_{i=1}^{n}\bar{f}_i \cdot F_i, \quad D_v = \sum_{i=1}^{n}\hat{e}_i \cdot E_i + \sum_{i=1}^{n}\hat{f}_i \cdot F_i$, where

$$\overline{e}_i = \sum_{t=1}^{5} x_{t1} c_{i-t+2}, \qquad \hat{e}_i = \sum_{t=1}^{5} x_{t2} c_{i-t+2}, \qquad \overline{f}_i = \sum_{t=1}^{5} x_{t3} c_{i-t+2}, \qquad \hat{f}_i = \sum_{t=1}^{5} x_{t4} c_{i-t+2},$$

where $c_\omega = \cos(2\pi\omega/n)$. All scalars x_{ij}'s in the above definitions depend on valance n only and can be derived from $M_{b,2}$ explicitly. To prove the existence of a common tangent plane at an extra-ordinary point, one needs to show that computation of the normal vector is independent of k (the ID of a face adjacent to an extra-ordinary point [1], which determines the order of the control points of a patch): $(\sum_{i=1}^{n} \overline{e}_i E_{i+k} + \sum_{i=1}^{n} \overline{f}_i F_{i+k}) \times (\sum_{i=1}^{n} \hat{e}_i E_{i+k} + \sum_{i=1}^{n} \hat{f}_i F_{i+k})$. To prove this, it is sufficient to show that $\sum \overline{e}_i E_{i+k} \times \sum \hat{e}_i E_{i+k}$ is independent of k. The other parts can be proved similarly.

Note $\sum_{i=1}^{n} \overline{e}_{i-k} E_i \times \sum_{j=1}^{n} \hat{e}_{j-k} E_j = \sum_{i \le j} (\overline{e}_{i-k} \hat{e}_{j-k} - \overline{e}_{j-k} \hat{e}_{i-k}) E_i \times E_j$. To prove the above expression is independent of k, we only need to prove $\overline{e}_{i-k} \hat{e}_{j-k} - \overline{e}_{j-k} \hat{e}_{i-k}$ is independent of k:

$$\overline{e}_{i-k} \hat{e}_{j-k} - \overline{e}_{j-k} \hat{e}_{i-k} = \sum_{1 \le s,t \le 5} x_{s1} x_{t2} (c_{i-k-s+2} c_{j-k-t+2} - c_{j-k-s+2} c_{i-k-t+2}) = \sum_{1 \le s,t \le 5} x_{s1} x_{t2} (c_{i-j-s+t} - c_{j-i-s+t})/2$$

which is independent of k. Hence all the patches sharing a common extra-ordinary point have the same normal vector at the extra-ordinary point. Therefore, there exists a common tangent plane at an extra-ordinary point.

When $\lambda_0 \ge \lambda_2$, it can be proved similarly that the resulting surface does not have a common tangent place [1]. In fact, Eq. (6) and Eq. (7) can be used for many other analytic purposes as well. For example, the curvature property at an extra-ordinary point can be explicitly analyzed using these two formulas [2].

Although most of these properties of CCSS around an extra-ordinary vertex are well known, an explicit parametrization of CCSS nevertheless makes the analyzing process much more simple and intuitive. Moreover, our results possibly can be used for studying other unknown properties of CCSS as well. For instance, we are investigating the integrability of a CCSS using the results in this paper.
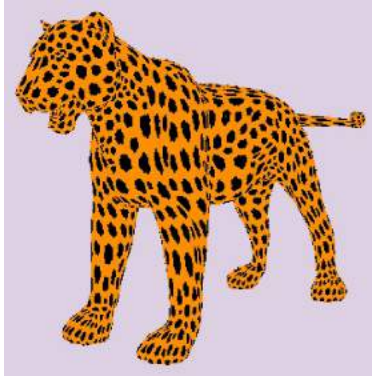
8. APPLICATIONS
### 8.1 Fast, Exact and Explicit Rendering
Eq. (6) not only gives us an explicit method to evaluate $S(u, v)$, but also a faster and convenient way to render $S(u, v)$. Note that $M_{b,j}$ depend on the valence of the extra-ordinary vertex only. They can be explicitly and analytically computed for every different valence. For a given valence, we only need to perform such calculation once, no matter how many patches in the mesh are with such a valence. Once the step sizes for u and v are given, we can calculate all $\emptyset_b(u_i, v_k)$ beforehand and store them in a look-up table. Therefore, the evaluation of $S(u, v)$ at each point (u_i, v_k) basically is just a multiplication of $\emptyset_b(u_i, v_k)$ and G only. All the examples shown in this paper are rendered using this method. An important difference between this approach and the previous approach [14] is that nothing need to be precomputed when our method is used, while the the Stam method [14] need to precompute a huge number of eigen basis functions and stored them in a file. In addition, the previous approach [14] was developed for special α_n and β_n only. Therefore, it cannot handle general eigen basis functions while we can calculate all the eigen basis functions explicitly with only a small overhead. The horse shown in Fig. 3(b) is rendered using this algorithm with all the positions and normals exactly computed, not approximated. Hence, the quality of the image is better than those generated through the subdivision process.
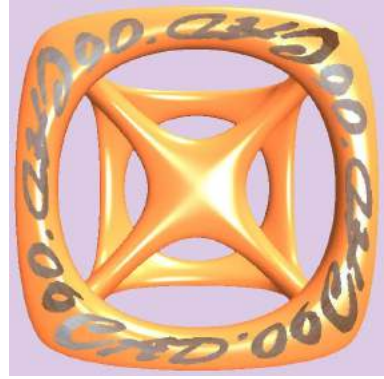
8.2 Generating Special Features
Eq. (6) can be used to render subdivision surfaces with special features. As we know, special features can be generated by properly arranging the control mesh. For instance, tripling a line in the control mesh generates a ridge or edge-like feature; tripling a control point generates a dart-like feature. One can get subdivision surfaces with complicated features and, consequently, complicated shape through this process. However, no matter how complicated the topology of the control mesh is, as long as it is a two-manifold (to satisfy the definition of a CCSS), Eq. (6) will always generate the correct result. An example of a CCSS with sharp edges, corners and several genera is shown in Fig. 3(i). The control mesh of the surface is shown in Fig. 3(h). Since the features are generated from parametrization of the control mesh directly, the result is better than those generated by Boolean operations.
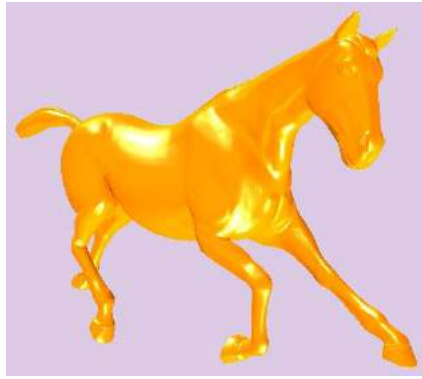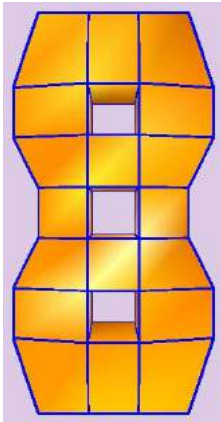
(a) Textured Surface (b) Exactly Evaluated Surface (c) Trimmed Surface
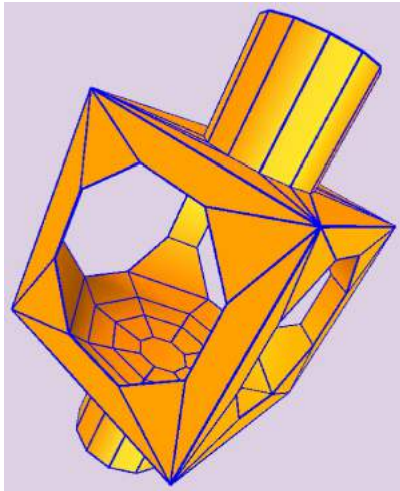
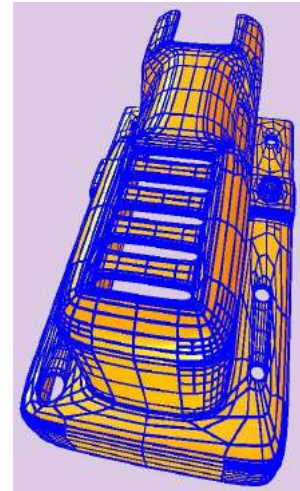(d) Given Mesh (e) Interpolation (f) Textured Surface (g) Boolean Operations

(h) Mesh with tripled edges (i) Surface with Special Features (j) Adaptive Rendering

Fig. 3. Application of parametric CCSS's.

8.3 Texture Mapping

Precise texture mapping on a CCSS is possible only if a proper parametric representation is available for each extra-ordinary patch. However, to implement texture mapping on a CCSS, one needs to divide the interior faces of the control mesh into regions such that each region is of a rectangular structure first. Such a division will be called a *regular division*. Once a regular division of the interior faces of the control mesh is available, one simply performs texture mapping on each of these regions using standard approaches. Examples of texture mapping on two subdivision surface represented objects are shown in Fig. 3(a) and Fig. 3(f). The regular division usually is not unique. Different divisions of the interior faces of the control mesh would lead to different texture outputs.

8.4 Surface Trimming

Surface trimming is another important application in computer graphics and CAD/CAM. The trimming loops are defined in the parameter space of the surface and iso-parametric lines in the parameter space are clipped against the trimming loops to have the trimmed regions removed. Hence, a global or local parametrization is necessary for precise and efficient rendering of a trimmed CCSS. In Fig. 3(c), a trimmed CCSS surface is shown. The trimmed regions are defined by the logo of the *2006 International CAD Conference*. The CCSS surface has four extra-ordinary vertices in the trimmed region, but partitioning of the control mesh is not required here because the surface is rendered on the basis of individual patches.

8.5 Adaptive Rendering

Adaptive rendering is a technique for fast rendering of complicated objects. The rendering process of a patch depends on its flatness. A flat patch will not be tessellated as densely as other patches. Adaptive rendering is not a problem with Eq. (6) because Eq. (6) is capable of generating any point of the surface required in the tessellation process. One thing we must keep in mind is that, in order to avoid crack, we must generate the same number of points on the shared boundary of adjacent faces. But we can generate any number of points, even zero, inside a patch. An example of adaptive rendering is shown in Fig. 3(j) where a ventilation control component is represented by a single CCSS. The flatness of a patch is determined by the maximum norm of the second order forward differences of its control points.

8.6 Interpolation

Performing exact interpolation on meshes with arbitrary topology has been done by many people [8, 19-22]. Given a control mesh the goal is to produce a smooth and visually pleasing surface whose shape matches the original data points or given normals in the given mesh exactly. Usually many constrains on the interpolating surface need to be considered when optimization is used. For example, in [8], some energy fairing constrains are taken into account in building a global system. Because there was not an available explicit parameterization, the fairing process appeared to be very complicated in [8]. However, with our explicit parameterization and evaluation, all kinds of constrains can be integrated into the global system. For example, Fig. 3(e) is the interpolating result of the mesh given in Fig. 3(d) using the first, second and third derivatives as constrains.

8.7 Boolean Operations

In solid modeling, an object is formed by performing Boolean operations on simpler objects or primitives. A CSG tree is used in recording the construction history of the object and is also used in the ray-casting process of the object. Surface-surface intersection and ray-surface intersection are the core operations in performing the Boolean operations and the ray-casting process. Each operation requires a parametrization of the surface to do the work. This is especially important for the in-on-out test. None of these is a problem with Eq. (6). Examples of performing Boolean operations on three cows are presented in Fig. 3(g). Performing Boolean operations on subdivision surfaces has been studied by Zorin [3]. The emphasis of their work is different though - they focus on construction of the approximating multiresolution surface for the result, instead of precise computation of the surface-surface intersection curves.

9. SUMMARY

New parametrization and evaluation techniques for extra-ordinary patches of CCSSs are presented in this paper. The parametrization is obtained by performing subdivision on a group of same-frequency point sets after a few linear transformations, not on the control vertices themselves directly. This results in a block diagonal matrix with constant size diagonal blocks (7×7) for the corresponding subdivision process. Consequently, eigen decomposition of the subdivision matrix is always possible and is simpler and more efficient. Besides, the new approach works for the general CCSSs, not just a special case. The evaluation process using this parametrization works for both extra-ordinary and regular CCSS patches. One thing has to be pointed out here. The exponent m in Eq. (6) can not be cancelled out.

This is because when λ_j is not a multiple of $1/2$, $m{-}1$ in K^{m-1} and $\lambda_j^{m-1} M_{b,j}$ does not cancel out. Hence, when $n \neq 4$, there does not exist a matrix M such that $S(u,v){=}W^T M\,G$.

10. ACKNOWLEDGEMENT

11. REFERENCES

[1]   Ball, A. A. and Storry, D. J. T., Conditions for tangent plane continuity over recursively generated B-spline surfaces, *ACM Transactions on Graphics*, Vol. 7, No. 2, 1988, pp 83-102.

[2]   Ball, A. A, and Storry, D, J, T,, An investigation of curvature variations over recursively generated B-spline surfaces, *ACM Transactions on Graphics*, Vol. 9, No. 4, 1990, pp 424-437.

[3]   Biermann, H., Kristjansson, D. and Zorin, D., Approximate Boolean operations on free-form solids, *Proceedings of SIGGRAPH*, 2001, pp 185-194.

[4]   Boier-Martin, I. and Zorin, D., Differentiable Parameterization of Catmull-Clark Subdivision Surfaces, *Eurographics Symposium on Geometry Processing*, 2004.

[5]   Catmull, E. and Clark, J., Recursively generated B-spline surfaces on arbitrary topological meshes, *Computer-Aided Design*, Vol. 10, No. 6, 1978, pp 350-355.

[6]   DeRose, T., Kass, M. and Truong, T., Subdivision Surfaces in Character Animation, *Proceedings of SIGGRAPH*, 1998, pp 85-94.

[7]   Lai, S. and Cheng, F., *Parametrization of General Catmull-Clark Subdivision Surfaces and its Applications*, http://www.cs.uky.edu/~cheng/publ.html, 2005.

[8]   Halstead, M., Kass, M. and DeRose, T., Efficient, fair interpolation using Catmull-Clark surfaces, *Proceedings of SIGGRAPH*, 1993, pp 35-44.

[9]   Litke, N., Levin, A. and Schroder, P., Trimming for Subdivision Surfaces, *Computer Aided Geometric Design*, Vol. 18, No. 5, 2001, pp 463-481.

[10]   Reif, U., A unified approach to subdivision algorithms near extraordinary vertices, *Computer Aided Geometric Design*, Vol. 12, No. 2, 1995, pp 153-174.

[11]   Peters, J. and Reif, U., Analysis of Algorithms Generalizing B-Spline Subdivision, *SIAM Journal of Numerical Analysis*, Vol. 35, No. 2, 1998, pp 728-748.

[12]   Sederberg, T. W., Zheng, J., Sewell, D. and Sabin, M., Non-uniform recursive subdivision surfaces, *Proceedings of SIGGRAPH*, 1998, pp 19-24.

[13]   Smith, J., Epps, D. and Sequin, C., *Exact Evaluation of Piecewise Smooth Catmull-Clark Surfaces Using Jordan Blocks*, http://www.cs.berkeley.edu/~jordans/pubs/, June, 2004.

[14]   Stam, J., Exact Evaluation of Catmull-Clark Subdivision Surfaces at Arbitrary Parameter Values, *Proceedings of SIGGRAPH*, 1998, pp 395-404.

[15]   Stam, J., Evaluation of Loop Subdivision Surfaces, *SIGGRAPH'99 Course Notes*, 1999.

[16]   Zorin, D. and Kristjansson, D., Evaluation of Piecewise Smooth Subdivision Surfaces, *The Visual Computer*, Vol. 18, Nos. 5/6, 2002, pp 299-315.

[17]   Zorin, D., Schroder, P. and Sweldens, W. Interactive Multiresolution Mesh Editing. *Proceedings of SIGGRAPH*, 1997, pp 259-268.

[18]   Warren, J. and Weimer, H., *Subdivision Methods for Geometric Design: A Constructive Approach*. ISBN: 1-55860-446-4, Academic Press, 2002.

[19]   Kobbelt, L., Interpolatory subdivision on open quadrilateral nets with arbitrary topology, *Computer Graphics Forum, Eurographics*, Vol. 15, 1996.

[20]   Zorin, D., Schroder, P. and Sweldens, W., Interpolating Subdivision for meshes with arbitrary topology, *ACM SIGGRAPH*, 1996, pp 189-192.

[21]   Dyn, N., Levin, D. and Gregory, J. A., A butterfly subdivision scheme for surface interpolation with tension control, *ACM Transactions on Graphics*, Vol. 9, No. 2, 1990, pp 160–169.

[22]   Nasri, A. H., Surface interpolation on irregular networks with normal conditions, *Computer Aided Geometric Design*, Vol. 8, 1991, pp 89–96.