Constructing Curvature-continuous Surfaces by Blending

Denis Zorin

New York University

Abstract

In this paper we describe an approach to the construction of curvature-continuous surfaces with arbitrary control meshes using subdivision. Using a simple modification of the widely used Loop subdivision algorithm we obtain perturbed surfaces which retain the overall shape and appearance of Loop subdivision surfaces but no longer have flat spots or curvature singularities at extraordinary vertices. Our method is computationally efficient and can be easily added to any existing subdivision code.

1. Introduction

Subdivision surfaces are well-established as a practical representation for geometric modeling with many useful properties. However, classical subdivision schemes like Loop and Catmull-Clark suffer from a number of problems: probably the best-known is the lack of C^2-continuity at the extraordinary vertices, i.e. vertices of the control mesh of valence different from 6 (Loop surfaces) and 4 (Catmull-Clark surfaces).

Several relatively simple solutions were proposed to this problem (e.g. [PU98]). However, ensuring formal C^2-continuity is not sufficient to solve all problems associated with absence of C^2-continuity. In particular, all simple approaches to making Loop or Catmull-Clark surfaces C^2-continuous at extraordinary vertices result in surfaces with *flat spots*: at surface points associated with an extraordinary vertex, the curvatures are forced to be zero. Careful rule tuning may make this artifact difficult to notice visually in most circumstances, but it will still exhibit itself for certain geometric configurations and certain types of lighting (e.g. reflection lines).

Even more importantly, C^2-continuity and absence of flat spots are needed for several types of numerical computation on surfaces. Examples include computation of curvature lines, which have singularities at flat spots and curvature singularities, computation of fairness functionals which require second derivatives and surface-surface intersection computations (e.g. one can construct examples of C^1 curves and surfaces intersecting in infinitely many isolated points).

The absence of flat spots for C^2 surfaces is more precisely described as *surface 2-flexibility*. Following Reif [Rei96], we say that a C^2-surface representation is 2-flexible, if for some C^2 parameterization any desired first and second derivatives can be obtained at a given point by a suitable choice of positions of the control points (see Section 4 for a precise definition). Surfaces with flat spots, or parametric points where the Gaussian curvature is always positive, are not flexible. On the other hand, if a surface is 2-flexible, the user is able to make the surface locally a paraboloid or a saddle with arbitrary orientation at any point. Flexibility is also related to surface approximation quality. If a surface has a flat spot, it cannot approximate C^2-surface in C^2-norm: the error remains constant.

In this paper, we introduce a new method for the construction of curvature-continuous flexible surfaces on arbitrary meshes, based on the idea of blending subdivision surfaces with locally defined surface patches. Our approach is a simple extension of common subdivision algorithms and can be easily implemented on the top of an existing subdivision framework. The appearance of the resulting surfaces is similar to the appearance of standard subdivision surfaces and have insignificantly higher computational cost. We were able to verify that resulting surfaces are flexible everywhere under certain assumptions are imposed on the control mesh.

Compared to a existing constructions of C^2 subdivision surfaces (Section 2), the distinguishing feature of our specific construction is that it remains very close to standard subdivision, while eliminating curvature discontinuity and flat-spot related problems and maintaining 2-flexibility away from extraordinary vertices. While we describe our construction for Loop surfaces, and our proofs are restricted to this case, the

extension to Catmull-Clark surfaces is straightforward and similar techniques can be used for analysis.

2. Previous Work

A large number of C^2 constructions for arbitrary meshes of various types were proposed over years. We mention some representative work. Hagen and Pottmann [HP89] C^2 interpolants of boundary position, tangent and curvature data are constructed. Gregory and Hahn [GH89] describe a C^2 hole-filling algorithm; Bohl and Reif [BR97] describe C^2- conditions on degenerate patches and how N patches can be joined at a point. C^2 spline surfaces on arbitrary meshes were constructed by Peters [Pet96] and higher order spline surfaces are described by Prautzsch in [Pra97]. More recently, various types of constructions based on polynomial patches were proposed in [Pet02], [Loo04] and [KP05].

C^2 subdivision algorithms based on standard schemes and with zero curvature at extraordinary vertices were proposed by Umlauf [PU98] and Biermann *et al.* [BLZ00].

The idea of obtaining smooth surfaces for arbitrary meshes using blending and appropriate local parameterizations, while known in geometric modeling for a long time (e.g. [GH89]), is used in more general form in the work on manifold-based surfaces [GH95, NG00, YZ04].

A closely related technique for subdivision surfaces was independently developed by Levin [Lev06].

The flexibility of resulting surfaces at arbitrary points is rarely addressed explicitly but, for many spline constructions, can be relatively easily inferred from the surface construction. For representations based on blending, a complete analysis is far more complex.

Despite the broad variety of options proposed in the research literature, the practice is dominated by non-C^2-continuous algorithms.

The difficulty of constructing a practical C^2-continuous surfaces appears to be in achieving the right tradeoff between mathematical properties (C^2-continuity and flexibility), visual quality, which can be captured by fairness measures, computational expense and the difficulty of implementation.

Compared to previous work, our main contribution is to propose a simple algorithm which can be added to an existing implementation of Loop subdivision with minimal effort and in most cases, yields surfaces closely approximating the standard Loop surfaces, yet curvature-continuous and 2-flexible everywhere.

The crucial ideas we build on are: obtaining smooth surfaces by blending in appropriate parameterization and using characteristic maps [Rei95] to obtain such parameterizations.

3. Overview

The basic idea of our construction is to blend a subdivision surface with parametric quadratic patches near extraordinary vertices. The quadratic patches are constructed from the control points in such a way that flexibility is guaranteed at the vertices.

For a given extraordinary vertex v, we use inverse of the *the characteristic map* to obtain local parameterization of the surface, which is C^2 away from v. The quadratic patch is defined as a function on to domain of the parameterization, i.e. the characteristic map image.

The blending basis function for the domain is taken to be the subdivision basis function corresponding to the extraordinary vertex, computed using a flat-spot modification of the Loop scheme.

Near the extraordinary vertex, the surface is blended with the quadratic patch using the blending function, so that the weight of the surface at v is zero. As we discuss below, this leads to C^2 surfaces flexible at vertices.

The distinguishing feature of the proposed construction is that three components of the construction (the surface itself, the characteristic map, and the blending function) can be computed using the same subdivision code, and the remaining component (the quadratic patch) is easy to evaluate.

4. Notation and terminology

To describe our construction and its properties in detail we briefly review the necessary notation and terms. We use boldface letters to denote 3D or 2D vectors.

Flexibility.

Definition 1 Let F be a parametric family of functions with values in $\mathbf{R}^n$ defined on a domain $D \subset \mathbf{R}^2$. We say that this family is *parametrically r-flexible* at a point $x \in D$, if for any given set of prescribed values of all partial derivatives $d_{ij}, i = 0 \ldots r, j = 0 \ldots i$ up to order r, there is a function $f(x_1, x_2) \in F$ with this set of derivatives at point $x \in D$:

$$\frac{\partial^i f}{\partial^j x_1 \partial^{i-j} x_2}(x) = d_{ij}$$

In particular, a function family is 2-flexible at x, if there is a function in the family with any prescribed values and prescribed first and second derivatives at x, which implies that it is possible to obtain arbitrary prescribed curvatures and curvature directions.

As explained below, subdivision defines a family (or more precisely a linear space) of functions $\sum \mathbf{p}_v B_v$ on a mesh, parameterized by the control points. For this family, *r*-flexibility means that we can choose the positions of the control points in such a way that at any fixed point x of the mesh we have prescribed partial derivative values up to order r,

with derivatives computed with respect to certain local parameterizations (characteristic map parameterizations).

Subdivision surfaces as functions on meshes. It is well known [Rei95] that common subdivision surfaces such as Loop and Catmull-Clark can be thought of as infinite collections of polynomial patches. The domains for these patches can be taken to be subtriangles or subquads associated with faces of the initial mesh. In particular, one can regard a patch of the subdivision surface corresponding to a ring of triangles adjacent to an extraordinary vertex to be a function on a regular k-gon U_k centered at $(0,0)$. While in the interior of each triangle, this function is C^2 for subdivision schemes extending C^2-continous splines. In general, one can only expect C^0 continuity between triangles. A different parameterization that we described below is needed to obtain C^2 on edges. Such parameterization is provided by characteristic maps.

Subdivision matrix and characteristic maps. A characteristic map is defined using the eigenstructure of the *subdivision matrix*, introduced in [DS78]. Consider a vertex v, and let p be the vector of control points in a neighborhood of the vertex; For the Loop subdivision scheme, we use all control points in a double ring of triangles around the vertex. From now on, we call the double ring of triangles around a vertex v the *2-neighborhood* of v; we call the single ring of triangles the *1-neighborhood* of v.

Let S be the $N \times N$ matrix of subdivision coefficients relating the vector of control points p^j on subdivision level j to the vector of control points p^{j+1} in a similar neighborhood on the next subdivision level. Many properties of the subdivision scheme can be deduced from the eigenstructure of the matrix. This is seen by decomposing the vector of control points p with respect to the eigenbasis $\{x^i\}$ of S, $i = 0..N-1$, assuming it exists:

$$p = \mathbf{a}_0 x^0 + \mathbf{a}_1 x^1 + \mathbf{a}_2 x^2 + \dots \quad (1)$$

where the multiplication of three-dimensional coefficients $\mathbf{a}_i$ with N-dimensional vectors x_i is understood in tensor-product sense and yields a $N \times 3$ matrix (the vector of 3d control points). We assume that the eigenvectors x^i are arranged in the order of non-increasing eigenvalues, and the first eigenvalue λ_0 is 1, which is required for convergence of subdivision. Furthermore, we assume that the corresponding eigenvector x^0 is the vector of ones necessary for affine invariance.

For control points in general position, the limit position of the center control point is $\mathbf{a}_0$ and the tangent directions at this position are $\mathbf{a}_1$ and $\mathbf{a}_2$.

Two subdominant eigenvectors x^1 and x^2 are used to construct the characteristic map for valence k. In the cases of interest to us, the corresponding eigenvalues are equal $\lambda_1 = \lambda_2 = \lambda$. The characteristic map Φ_k is defined as the limit function of a subdivision for a 2D mesh which is constructed as follows. The mesh consists of two rings of triangles around a vertex of valence k. The coordinates of the vertices of the initial mesh are taken to be the components of eigenvectors x^1 and x^2 respectively, $\mathbf{p}_j = (x_j^1, x_j^2)$, $j = 0 \dots N - 1$. The 1-neighborhood of the central vertex is typically a regular k-gon, or can be mapped to one by a piecewise-linear mapping. The characteristic map Φ_k is the limit function with values in $\mathbf{R}^2$ generated by subdivision from this initial mesh, and restricted to the regular k-gon (Figure 1).

The following property is most important for us: the characteristic maps for all valences are injective. Moreover, F is the parameterization of subdivision surface over a regular k-gon described above; the composition $F \circ \Phi_k^{-1}$ is C^2-continuous if the scheme is C^2-continuous in the regular case.
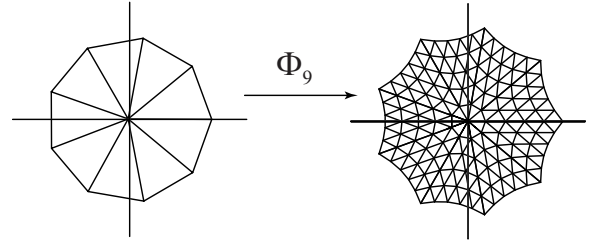

Figure 1: *Characteristic map of the Loop subdivision scheme for valence $k = 9$. On the left, a piecewise linear approximation to the image of the map is shown.*

More generally, the subdivision surface can be represented as a linear combination of *eigenbasis functions* f_i i.e. functions obtained from the eigenvectors x_i by subdivision (this amounts to a change of basis in (1)). An eigenbasis f_i satisfies $f_i(t/2) = \lambda_i f(t)$ on the regular k-gon and its characteristic map reparameterization satisfies $(f_i \circ \Phi_k^{-1})(\lambda x) = \lambda_i (f_i \circ \Phi_k^{-1})(x)$. It easily follows from this formula [Zor00] that the function f_i changes as $|x|^\alpha$ near $x = (0,0)$ with $\alpha = \log \lambda_i / \log \lambda$.

Subdivision scheme. We use two versions of Loop subdivision. The first is a commonly used version of Loop subdivision with the vertex rule

$$p_0^{j+1} = (5/8)p_0^j + (3/8k)\sum_{i=1}^{k} p_i^j \quad (2)$$

Note that a special rule is normally necessary for $k = 3$ as in this case the resulting surface is not C^1-continuous for the rule (2). We use the same rule for $k = 3$ as the smoothness of subdivision surface at the extraordinary vertex does not affect the smoothness of the blended surface.

A modified version of this scheme is used to compute the

blending function used in our construction. We use a special case of the scheme from [BLZ00]. On the first subdivision step, values for vertices immediately adjacent to an extraordinary vertex are computed using

$$p^1 = \mathbf{a}_0 + \lambda \mathbf{a}_1 x_1 + \mathbf{a}_2 \lambda x_2 \tag{3}$$

Note that this forces the control values adjacent to the extraordinary vertex to be in the same plane, which turns out to be sufficient to ensure that the limit function is C^2 with zero second derivatives in the characteristic map parameterization. After the first step, the standard rules are used.

Also note that eigenvectors of the subdivision matrix are not affected by the modification. The only change to the eigenvalues is that for a subset of eigenbasis functions the coefficients are set to zero. These eigenbasis functions include all functions which are obtained as linear combinations of basis functions corresponding to projected control points in the one-ring, excluding f_0, f_1 and f_2.

5. Blending local patches with subdivision surfaces

First, we describe how to blend a quadratic patch $\mathbf{Q}(t)$ associated with a vertex v with the subdivision surface produced by the Loop scheme. We assume that the patch $\mathbf{Q}(t)$ is defined as a smooth function from the plane into $\mathbf{R}^3$, so t is a point on the plane. A specific approach to computing local quadratic patches is described in Section 5.1.

We start by applying one step of refinement to the subdivision surface, to obtain a new control mesh M^1. The 1-neighborhoods of the vertices of M in M^1 share only isolated points (Figure 2).

The blending is restricted to the 1-neighborhoods of extraordinary vertices in M^1 i.e. at every point of the surface at most one quadratic patch is blended with the surface.

As we have discussed, a subdivision surface is a function defined on the control mesh, $\mathbf{F}(x) = \sum_v \mathbf{p}_v B_v(x)$, where $\mathbf{p}_v$ are the control points, $B_v(x)$ are the basis functions of subdivision.
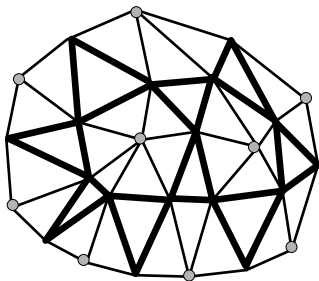

Figure 2: *1-neighborhoods of the vertices of the initial mesh M in the once-refined mesh M^1.*

We construct the blending function B_k^2 as follows. Take a

mesh R_k with a single central extraordinary vertex v of valence k, and containing a double ring of vertices around v. Subdivide this mesh once to obtain R_k^1. Assign the value 1 to v and zeros to all other vertices and apply subdivision to these values. This yields a scalar basis function defined on a 2-neighborhood of v in R_k^1, i.e. on 1-neighborhood of v in R_k, which we identify with the regular k-gon. We rescale function values so that the value at the extraordinary vertex is 1. The resulting function $B_k^2 : U_k \to \mathbf{R}$ is the blending function we use.

Let Φ_k be the characteristic map as defined in Section 4.

We use the following formula for blending the patch with the surface on the regular k-gon U_k:

$$\mathbf{F}^{\text{blended}}(x) = \left(1 - B_k^2(x)\right)\mathbf{F}(x) + B_k^2(x)\mathbf{Q}(\Phi_k(x)) \tag{4}$$

Our new surface is a blend of the old surface $\mathbf{F}(x)$ and the new locally defined patch $\mathbf{Q}(t)$ with the contribution of $\mathbf{Q}(t)$ reducing to zero at the boundary of 1-neighborhood of v in M^1, i.e. half-way to the adjacent vertex in the original mesh M. As we will see, all three components required to compute $\mathbf{F}^{\text{blended}}$ can be easily evaluated.

The proof of C^2-continuity at extraordinary vertices. As it was discussed in Section 4, if Φ_k is the characteristic map, $\mathbf{F} \circ \Phi_k^{-1}$ is differentiable.

We reparameterize the blended surface $\mathbf{F}^{\text{blended}}(x)$ over a domain in the plane (the image of the characteristic map),

$$\begin{aligned} \mathbf{F}^{\text{blended}}(\Phi_k^{-1}(t)) = &(1 - B_k^2(\Phi_k^{-1}(t)))\mathbf{F}(\Phi_k^{-1}(t)) + \\ &B_k^2(\Phi_k^{-1}(t))\mathbf{Q}(t) \end{aligned} \tag{5}$$

For standard Loop subdivision near extraordinary vertices the second derivatives of $\mathbf{F} \circ \Phi_k^{-1}$ grow no faster than $t^{\log \lambda_2 / \log \lambda - 2}$, where λ_2 is the next largest eigenvalue after the subdominant eigenvalue λ [Zor00]. Specific flat-spot subdivision rules described in Section 4 ensure that the only eigenbasis functions that contribute to the blending function derivative behavior at zero are the eigenbasis functions corresponding to control points outside the one-ring of the vertex. The eigenvalues of these functions are known to be less than $1/8$, so the decay rate of these functions near the extraordinary point is faster than $|x|^{\alpha_1}$ for $\alpha_1 = \log(1/8)/\log \lambda$ in the characteristic map parameterization. For the surface itself the decay rate can be estimated to be no slower than $\alpha_2 = \log \lambda_2 / \log \lambda$. As a consequence the increase rate of second derivatives does not exceed $|x|^\alpha$ for $\alpha = \alpha_2 - 2$, if $\lambda_2 > \lambda^2$.

Using these observations and the product rule for second derivatives one concludes that the rate of change of second derivatives of $(1 - B_k^2(x))\mathbf{F}(x)$ in the characteristic map

reparametrization is $\alpha_1 + \alpha_2 - 2$, which can be verified to be positive if $\lambda_2 < 8\lambda^2$, which can be shown to hold for any valence for Loop subdivision.

The blending function $B_k^2 \circ \Phi_k^{-1}$ is C^2-continuous by construction.

We conclude that the characteristic map parametrization of the blended surface is C^2-continuous because all included functions are C^2-continuous. Moreover, all first and second derivatives of the first part at $(0,0)$ are zero by construction, and the derivatives of the second part coincide with the derivatives of $Q(t)$; thus we have complete control over surface flexibility through the choice of $Q(t)$.

5.1. Local quadratic patches

The patch $\mathbf{Q}(t)$ for an extraordinary vertex v is constructed as a function in the plane, with values in $\mathbf{R}^3$. The idea of the construction is to find a quadratic patch that follows the local shape of the mesh; at the same time, we try to make our surfaces as similar as possible to the surfaces generated using subdivision.

Thus we use the formulas for the limit positions of Loop subdivision for $\mathbf{Q}(0)$. Remarkably, the rest of the coefficients can be found using least-squares fit, and the resulting formulas for the coefficients for the first derivatives coincide with the standard formulas for tangents to the Loop subdivision surfaces.

The coefficients can be obtained as follows. Let (r, φ) be the polar coordinates in the plane. Then the second-order approximation to the surface can be written as

$$\mathbf{Q}(r, \varphi) = \mathbf{b}_0 + (\mathbf{b}_{11} \cos \varphi + \mathbf{b}_{12} \sin \varphi)r +$$
$$(\mathbf{b}_{20} + \mathbf{b}_{21} \cos 2\varphi + \mathbf{b}_{22} \sin 2\varphi)r^2$$

We assume that $\mathbf{b}_0$ is computed using the formula for the limit positions of a control point for Loop subdivision, that is, $\mathbf{b}_0 = \mathbf{p}_0/2 + (1/2k)\sum_i \mathbf{p}_i$, for $k \neq 3$, $a_0 = 2\mathbf{p}_0/5 + \sum_i \mathbf{p}_i/5$. We determine the other five coefficients by fitting a quadric with $\mathbf{b}_0 = 0$ to the shifted control points $\mathbf{p}_i - \mathbf{p}_0$, $i = 0 \ldots k - 1$.

A simple calculation shows that the least squares fit to k points of the 1-neighborhood $\mathbf{p}_1 - \mathbf{p}_0 \ldots \mathbf{p}_k - \mathbf{p}_0$ assumed to be values at $(\cos(2\pi i/k), \sin(2\pi i/k))$, $i = 0..k - 1$, leads to

$$\mathbf{b}_{11} = \frac{2}{k} \sum_i \mathbf{p}_i \cos \frac{2\pi i}{k}; \quad \mathbf{b}_{12} = \frac{2}{k} \sum_i \mathbf{p}_i \sin \frac{2\pi i}{k}$$

$$\mathbf{b}_{20} = -\mathbf{p}_0 + \frac{1}{k} \sum_i \mathbf{p}_i$$

$$\mathbf{b}_{21} = \frac{2}{k} \sum_i \mathbf{p}_i \cos \frac{4\pi i}{k}; \quad \mathbf{b}_{22} = \frac{2}{k} \sum_i \mathbf{p}_i \sin \frac{4\pi i}{k}$$

Note that the formulas for $\mathbf{b}_{11}$ and $\mathbf{b}_{21}$ coincide with the standard formulas for the tangents to the Loop subdivision surface, and $\mathbf{b}_{20}, \mathbf{b}_{21}, \mathbf{b}_{22}$, with appropriate variable changes, produce second derivatives in the regular case. As a result, we obtain a set of simple rules for computing the coefficients of an approximating quadratic surface which can be used as function $\mathbf{Q}(t)$ in (4). A similar construction can be used for the boundary, but we do not consider it here. Example set of quadratic patches is shown in Figure 3.
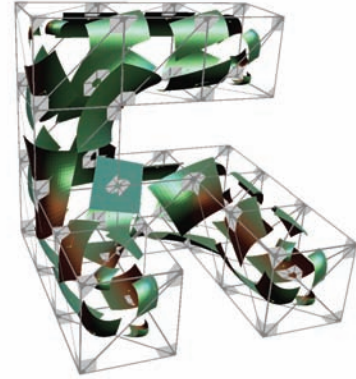

Figure 3: *Local quadratic patches are used to approximate a surface near extraordinary points.*

Flexibility. It is easy to show that for valences $k \geq 5$, the patches are parametrically 2-flexible: for any specified first and second derivatives, we can solve for patch coefficients $\mathbf{b}_i$ yielding these derivatives. For $k = 3, 4$, the patches are not 2-flexible, as the number of control points is less than the total number of the derivatives of order ≤ 2.

To obtain surfaces that are parametrically flexible for all valences, we need to use special rules for $k = 3, 4$. Such rules can be obtained in a similar manner, but require using triangles outside the 1-neighborhood of the vertex. We use the standard masks for $\mathbf{b}_0$ (the limit position) and $\mathbf{b}_{11}, \mathbf{b}_{12}$ (tangents). In both cases $\mathbf{b}_{20}$ can be computed using a single ring of vertices and expressions specified above. In the case of $k = 4$, the mask for $\mathbf{b}_{21}$ does not require changes either. In the remaining cases (both "saddle" coefficients $\mathbf{b}_{21}$ and $\mathbf{b}_{22}$ for $k = 3$ and one of "saddle" coefficients $\mathbf{b}_{22}$ for $k = 4$) additional vertices need to be used. We obtain these coefficients using a least squares fit to the single ring of control points around the vertex augmented by k triangles outside the ring. The resulting masks are shown in Figure 4.

Even if these special masks for valences 3 and 4 are used, the surfaces may still be not parametrically 2-flexible for some meshes. There are four small closed meshes for which the
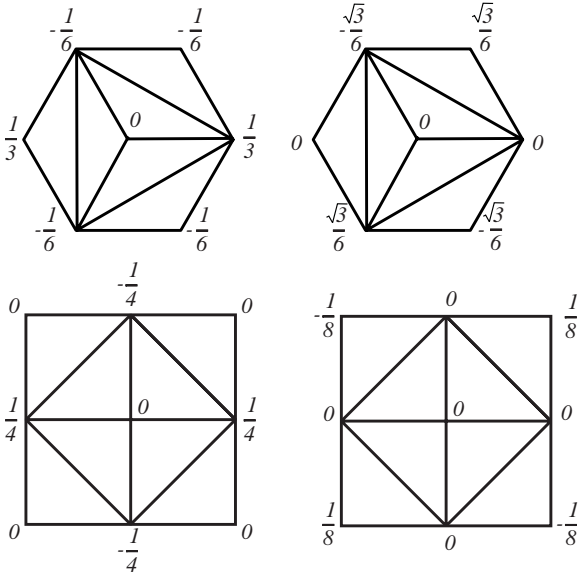
Figure 4: *Masks for computing coefficients* $\mathbf{b}_{21}$ *and* $\mathbf{b}_{22}$ *for valences* $k = 3$ *and* $k = 4$.

surface is not parametrically 2-flexible, including the octahedron and tetrahedron (Figure 5, *b,c,f,g*), due to insufficient total number of degrees of freedom in the double ring.

In addition, the surface is not parametrically flexible for meshes containing the three configurations *a,d,e* shown in Figure 5. While configuration *e* is somewhat unusual, the configuration *d* occurs, for example, if a pyramid is joined with its reflected image at the base. There does not appear to be a simple solution to this problem in our framework. While it is unlikely to be an issue for tetrahedron and octahedron, it is likely that a larger mesh would contain a neighborhood shown in Figure 5. It should be noted that the degree of inflexibility is relatively small in this case: a mixed second derivative is constrained to be zero.

5.2. Algorithm

Finally we summarize the basic algorithm for computing our surfaces. If a direct evaluation routine of the type described in [Sta98] is available, than the implementation amounts to defining a set of control meshes for the characteristic map and blending functions, as described in more detail below, and calling this routine to evaluate (4).

The output of the algorithm is a mesh approximating the surface described by (4) after *n* refinement steps. All terms in (4) are computed using subdivision modified Loop subdivision with zero curvature.

One refinement step is performed first; the refined mesh is M^1. Each triangle of the refined mesh either has a single ex-

traordinary vertex (valence not equal to 6) or all its vertices are regular. All control points that are inserted in triangles with regular vertices are computed using standard subdivision rules.

For each triangle *T* which has an extraordinary vertex, we can characterize any vertex obtained by refining this triangle by its barycentric coordinates (u,v,w). The barycentric coordinates have the form $(i/2^n, j/2^n, 1 - (i+j)/2^n)$, because all new vertices are inserted using midpoint subdivision. We always choose the coordinates in such a way that the weight *w* corresponds to the single extraordinary vertex in the triangle; the last coordinate *w* can be dropped, as $u + v + w = 1$. Finally, assuming *n* fixed, each vertex is identified by the pair of indices (i, j). For most common representations of the subdivision meshes the indices (i, j) or a some modified form of these indices is readily available. When evaluating (4) we are interested only in values at the vertices obtained after *n* subdivision steps, which can be enumerated using the indices $(i, j), i, j = 0 \ldots 2^{n-1}, i + j < 2^n$, as above, with $(0,0)$ corresponding to the extraordinary vertex.

To compute an approximation to the surface after *n* subdivision steps, the algorithm proceeds as follows.

Precomputation.

1. For each extraordinary vertex, we precompute and store the coefficients of the quadric associated with the extraordinary vertex applying the coefficients of Section 5.1 to a ring of vertices around the extraordinary vertices.

2. For each valence *k* present in the mesh, we precompute the limit values of the characteristic map Φ_k and $B_k^2(x)$ at vertices (i, j) at the subdivision level *n* for one triangular sector (all other values can be obtained by applying an appropriate rotation). For each valence *k*, we create two small meshes, both consisting of 2 rings of triangles around an extraordinary vertex of valence *k*. To the vertices of the mesh used to compute B_k^2 we assign scalar initial values. To obtain B_k^2, we refine twice first, then set all values to zero except the value in the center, which is set to 2, so that the limit value in the center is 1. The second mesh, used to compute Φ_k, is initialized to values from $\mathbf{R}^2$. The two components of each value are the components of the subdominant eigenvectors for valence *k*. As we care only about the values of the functions on the first triangle T_0 of the *k*-gon, we need to refine only this triangle. After refining *n* times and using the standard limit rules, we obtain the values $\Phi_k(i, j)$ and $B_k^2(i, j)$ at all vertices of *n* times refined triangle T_0.

Evaluation. We first subdivide the mesh *n* times using Loop rules, and evaluate the limit position for each vertex. Fix a triangle $T = (v_0, v_1, v_2)$ of M^1 with an extraordinary vertex v_0.

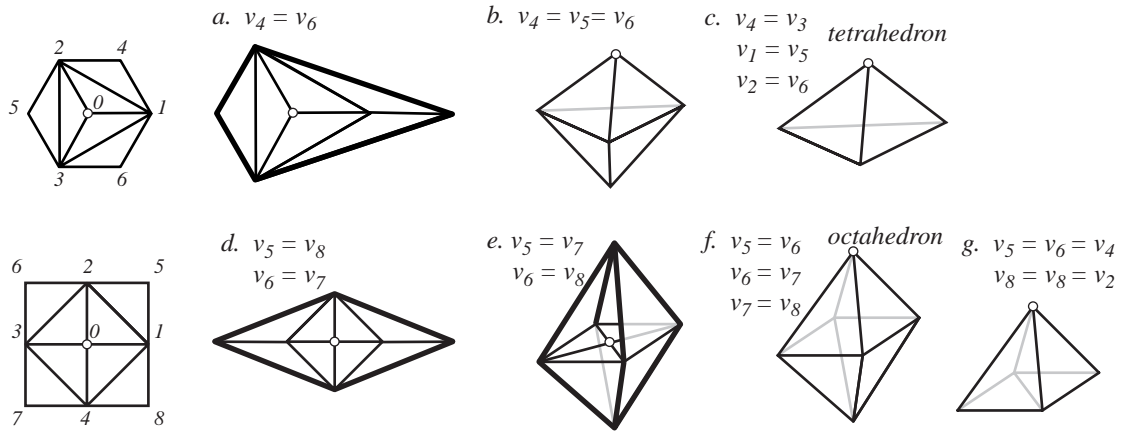For each vertex inserted into a triangle of M^1 adja-

Figure 5: *Inflexible meshes. Upper row: meshes inflexible at a vertex of valence* 3. *Lower row: meshes inflexible at a vertex of valence* 4. *For each mesh, the circle indicates the vertex where the mesh is not 2-flexible. Naming convention for vertices is shown on the left. Each inflexible mesh is obtained by identifying some vertices, as indicated by the equations. Only three meshes are open, that is, have boundary edges (shown as thick) and can be submeshes of larger meshes. The remaining meshes are closed. Invisible edges are shown in gray.*

cent to an extraordinary vertex, we determine the indices (i, j) and compute the final value $(1 - B_k^2(i,j))\mathbf{F}(i,j) + \mathbf{Q}(\Phi_k(i,j))B_k^2(i,j)$.

Note that computation of B_k^2 and Φ_k have to be performed only once per valence, that is, they have little impact on the performance of the algorithm; the only additional expense for each vertex is the lookup of the values $\Phi_k(i,j)$ and $B_k^2(i,j)$, and computation the linear combination (4).

A simple implementation of the algorithm required less than 500 lines of code on the top of an existing subdivision library.

5.3. Surface Quality

As it can be seen in Figure 7, for common meshes the difference between surface appearance is difficult to see. However, in some cases the difference can be significant (Figure 6). We have chosen to compare surface appearance for standard Loop surfaces and surfaces obtained by blending using a saddle-shaped control mesh. The reason for this is that for valences higher than six surfaces corresponding to such control meshes have in some sense least curvature continuity. As it is discussed in detail in [Zor00] the lack of curvature continuity is due to the fact that certain eigenvalues of the subdivision matrix have magnitude larger than λ^2 where λ is the subdominant eigenvalue. For the Loop scheme, the largest among these eigenvalues is the eigenvalue $3/8 + (1/4)\cos 4\pi/k$; the corresponding eigenvector is v_{21}. A saddle-like arrangement of control points has the largest magnitude of the corresponding coefficient $\mathbf{b}_{12}$ in the eigenbasis decomposition, which results in pronounced vi-

olation of curvature continuity especially for high valences. For example in Figure 6 for valence 20 one can see that the surface develops a visible kink. At the same time for valences close to 6 no artifacts are visible. However, the curvature still diverges for valences higher than 6, which causes numerical problems for algorithms such as surface-surface intersection and geodesic tracing. In contrast, the overall shape of blended surfaces is not much affected by valence and the curvatures converge smoothly to a limit value (Figure 6). It appears that the surfaces are slightly bumpier away from the extraordinary points but on the models that we have considered the effect was not very pronounced. The fact that the Gaussian curvature for the saddle becomes positive is also undesirable and is likely to be correlated with bumpiness. The visible ripple artifacts for valence 20 are due to the artifacts present in the Loop basis function used for blending. While these artifacts are absent when standard Loop is applied to the saddle eigenvector configuration, they immediately appear once other eigenvectors are present as it is the case for any real surface. One can hope that a blending function with no ripple artifacts would produce even better results, but it is not clear how to construct such functions with k-gonal support or if it is possible to use different support shapes without introducing a different type of artifact (in [Lev06] a different blending function is explored.)

6. Analysis of Flexibility

Parametric 2-flexibility at extraordinary vertices with respect to the characteristic map parameterization was easily established by construction.

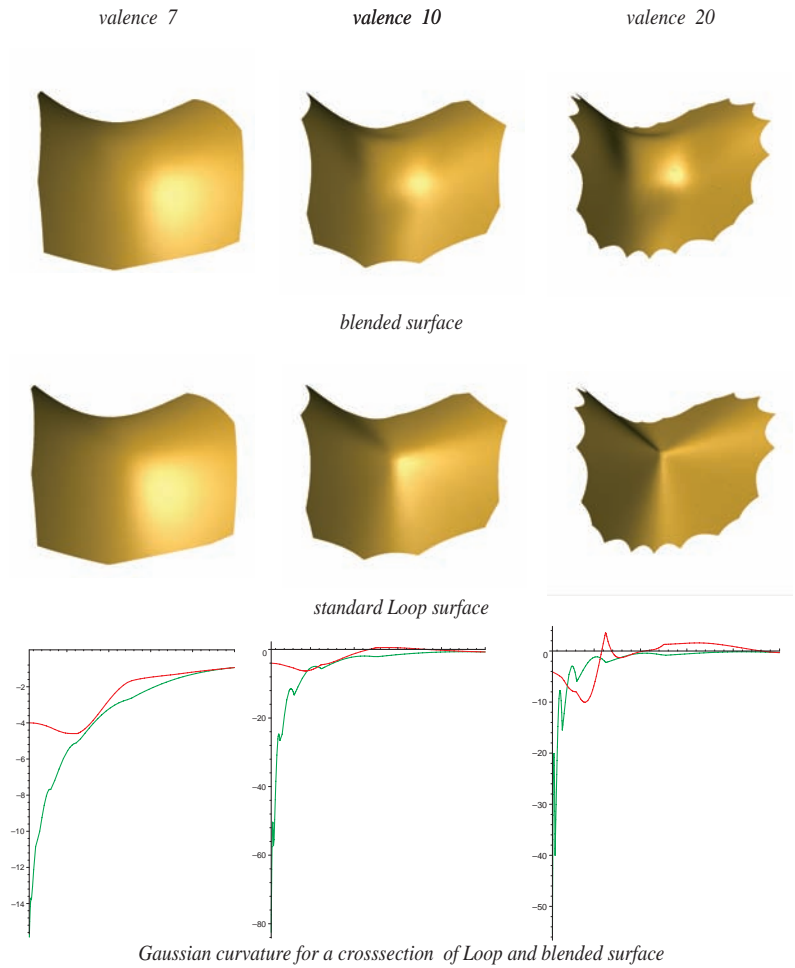However our construction does not *a priori* guarantee that

valence 7   **valence 10**   *valence 20*

blended surface

standard Loop surface

Gaussian curvature for a crosssection of Loop and blended surface

Figure 6: *Behavior of the standard Loop surface and the blended surface for valences 7,10 and 20. The range for curvature plots is 0.01 to 0.5 measured barycentric coordinates along a line passing through the origin. The red curve is the curvature for the blended surface and the green curve is the curvature for the red surface.*
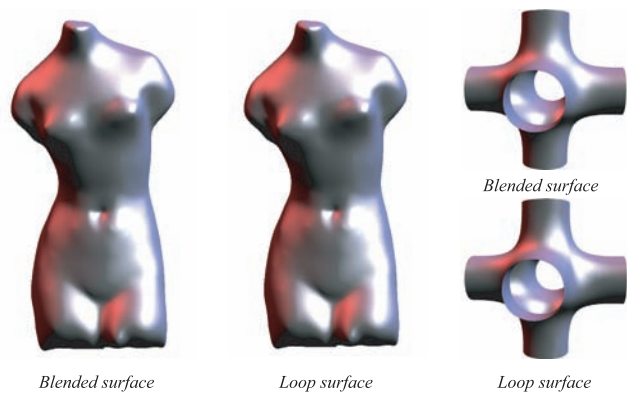

Blended surface

Blended surface   *Loop surface*   *Loop surface*

Figure 7: *Left: A slight difference exists but hard to identify visually for this model. Right: the difference in the shape of highlights for simple models is more apparent. We reiterate that our goal not as much improve surface fairness but retain it and obtain surfaces with better mathematical properties in a simple way.*

the resulting surfaces are flexible away from the vertices of the top-level mesh. While not very probable, it is possible that as a result of blending we introduce inflexible points at locations away from the extraordinary vertex. Thus, to establish that the proposed approach is guaranteed to result in 2-flexible everywhere for a sufficiently broad class of meshes further analysis is required. The analysis of flexibility of blended at points away from extraordinary points, while relatively straightforward conceptually proved to be difficult technically. Here we outline the idea of the proof which extensively uses computer algebra. More details, and relevant computer algebra code is presented as a separate report. *A note to the reviewers: the report is included in the paper submission.*

We are able to show that the surfaces produced by the scheme are flexible on closed control meshes with two additional constraints imposed:

C1: Extraordinary vertex separation. No two extraordinary vertices are adjacent.

C2: Simple neighborhood topology. All $k + 6$ vertices in the 1-neighborhood of any triangle which has a vertex of valence $k \neq 6$ are distinct.

The argument can be also easily extended to the case when any two adjacent extraordinary vertices both have valence higher than six.

The case of meshes with adjacent low-valence extraordinary vertices presents considerable difficulty. In some cases, the resulting surfaces cannot be flexible as the number of basis functions with support overlapping some of the points of the surface can be less than six, the minimal number required for flexibility. There are few configurations like this however, and in most other case it is highly likely that the resulting surfaces are flexible. A brute-force analysis would require analyzing a large number of local configurations, computing the explicit piecewise polynomial representation of the limit surface for each. We leave such analysis for our scheme and other C^2 constructions which are potentially 2-flexible as future work.

As we analyze parametric 2-flexibility, analysis can be performed for each coordinate separately; therefore it is sufficient to consider scalar functions defined by subdivision.

Theorem 1 For a closed mesh M satisfying constraints C1 and C2, the blended surface defined by (4) is parametrically 2-flexible at any point of M.

Outline of the proof. For the vertices of the control mesh, flexibility immediately follows from the surface definition. We need to prove 2-flexibility for all other points of the surface. If any two extraordinary vertices are separated by regular vertices, we only need to consider triangles which have a single extraordinary vertex.

Next, consider the part of the surface that is defined on the

1-neighborhood of an extraordinary vertex. This part of the surface vertex depends on a double ring of control points around this vertex. Rather than proving that one can choose the positions of all of these control points to obtain the desired result, we use a set of 6 coefficients in the eigenbasis decompositions as degrees of freedom, i.e. we prove that for some choice of these coefficients we can obtain any desired first and second derivatives at any given point with barycentric coordinates (u, v, w) in a triangle adjacent to the extraordinary vertex (one can easily show that any set of these coefficients can be obtained using a suitable combination of control points).

As we have already mentioned, the surfaces we are interested in can be evaluated using Stam's algorithm at any point (u, v, w), with $u + v + w = 1$. Recall that using this algorithm, for a point (u, v, w), we evaluate the surface as a value of a quartic box spline patch with control points expressed as functions of control points of the subdivision surface and subdivision level i depending on (u, v, w). More specifically, the control points of this patch are linear combinations of control points with coefficients of the form $C\lambda_j^i$, where λ_j are eigenvalues of the subdivision matrix. It turns out that in the specific case that we consider a variable change simplifies the expressions for the surface at an arbitrary location to a polynomial $F(u, v, w, \varepsilon, c)$ in 5 variables u, v, w, ε, c, where $c = \cos(\pi/k)$, k is the valence, $\varepsilon = 2^{-i}$ is the subdivision level, with coefficients linearly depending on a_{ij}, $0 < j < i \leq 2$, and independent of k (the dependence on valence is completely captured by c).

Differentiating this polynomial with respect to u and v to obtain 6 derivatives of order ≤ 2; and prescribing the values of these derivatives yields a system of 6 equations in 6 variables a_{ij}, with coefficients which are polynomials in u, v, w, ε and c. The system has a solution whenever its determinant is not zero. As all coefficients are polynomials, the determinant is also a polynomial in the same variables; the ranges of the variables are $0 \leq u \leq 1$, $0 \leq v \leq 1$, $0 \leq u + v \leq 1$, $0 < c < 1$ (for $k \geq 5$), $0 < \varepsilon \leq 1$. To prove flexibility it is sufficient to show that this polynomial is greater than a constant C on the domain defined by listed inequalities. We achieve this by converting it to Bezier form and verifying that all coefficients are positive. The case $k = 3$ is considered in the same way but separately, with one less variable.

7. Conclusions and Future Work

The method that we have described easily extends to other types (e.g. Catmull-Clark) or higher-smoothness subdivision surfaces, e.g. if the subdivision rule is C^k in the regular case, the technique can yield C^k surfaces. However, flexibility away from extraordinary vertices needs to be verified separately in each case.

The choice of subdivision basis functions as blending functions in this paper was primarily motivated by considera-

tions of simplicity and efficiency, as well as the possibility of proving 2-flexibility away from extraordinary points. In general, the construction of blending functions need not be the same as this construction. Moreover, one can use blending functions to combine local surface patches of different types.

While proposed surfaces are C^2 and flexible, they still do not allow exact reproduction of certain important simple shapes. For example, it would be useful to be able to reproduce a sphere without seams, a modeling task, which, to the best of our knowledge, cannot be achieved by any existing parametric surface representation (a NURBS sphere has a seam), except trigonometric schemes proposed in [MWW01].

References

[BLZ00] BIERMANN H., LEVIN A., ZORIN D.: Piecewise smooth subdivision surfaces with normal control. In *SIGGRAPH 2000 Conference Proceedings* (2000), pp. 113–120.

[BR97] BOHL H., REIF U.: Degenerate bézier patches with continuous curvature. *Comput. Aided Geom. Des. 14*, 8 (1997), 749–761.

[DS78] DOO D., SABIN M.: Analysis of the behaviour of recursive division surfaces near extraordinary points. *Computer Aided Design 10*, 6 (1978), 356–360.

[GH89] GREGORY J. A., HAHN J. M.: A C^2 polygonal surface patch. *Comput. Aided Geom. Design 6*, 1 (1989), 69–75.

[GH95] GRIMM C. M., HUGHES J. F.: Modeling surfaces of arbitrary topology using manifolds. *Proceedings of SIGGRAPH 95* (August 1995), 359–368. ISBN 0-201-84776-0. Held in Los Angeles, California.

[HP89] HAGEN H., POTTMANN H.: Curvature continuous triangular interpolants. In *Mathematical methods in computer aided geometric design (Oslo, 1988)*. Academic Press, Boston, MA, 1989, pp. 373–384.

[KP05] KARČIAUSKAS K., PETERS J.: Polynomial C^2 spline surfaces guided by rational multisided patches. In *Computational methods for algebraic spline surfaces*. Springer, Berlin, 2005, pp. 119–134.

[Lev06] LEVIN A.: Modified subdivision surfaces with continuous curvature. In *Proceedings of SIGGRAPH 2006* (2006). to appear.

[Loo04] LOOP C.: Second order smoothness over extraordinary vertices. In *SGP '04: Proceedings of the 2004 Eurographics/ACM SIGGRAPH symposium on Geometry processing* (New York, NY, USA, 2004), ACM Press, pp. 165–174.

[MWW01] MORIN G., WARREN J., WEIMER H.: A subdivision scheme for surfaces of revolution. *Computer Aided Geometric Design 18*, 5 (June 2001), 483–502. ISSN 0167-8396.

[NG00] NAVAU J. C., GARCIA N. P.: Modeling surfaces from meshes of arbitrary topology. *Computer Aided Geometric Design 17*, 7 (2000), 643–671.

[Pet96] PETERS J.: Curvature continuous spline surfaces over irregular meshes. *Computer-Aided Geometric Design 13*, 2 (Feb 1996), 101–131.

[Pet02] PETERS J.: C^2 free-form surfaces of degree $(3,5)$. *Comput. Aided Geom. Design 19*, 2 (2002), 113–126.

[Pra97] PRAUTZSCH H.: Freeform splines. *Comput. Aided Geom. Design 14*, 3 (1997), 201–206.

[PU98] PRAUTZSCH H., UMLAUF G.: A G^2-subdivision algorithm. In *Geometric Modelling, Computing Suppl. 13*, Farin G., Bieri H., Brunnet G.,, DeRose T., (Eds.). Springer-Verlag, 1998, pp. 217–224.

[Rei95] REIF U.: A unified approach to subdivision algorithms near extraordinary points. *Computer-Aided Geometric Design 12* (1995), 153–174.

[Rei96] REIF U.: A degree estimate for polynomial subdivision surfaces of higher regularity. *Proc. Amer. Math. Soc. 124* (1996), 2167–2174.

[Sta98] STAM J.: Exact evaluation of catmull-clark subdivision surfaces at arbitrary parameter values. In *SIGGRAPH '98: Proceedings of the 25th annual conference on Computer graphics and interactive techniques* (New York, NY, USA, 1998), ACM Press, pp. 395–404.

[YZ04] YING L., ZORIN D.: A simple manifold-based construction of surfaces of arbitrary smoothness. *ACM Trans. Graph. 23*, 3 (2004), 271–275.

[Zor00] ZORIN D.: Smoothness of subdivision on irregular meshes. *Constructive Approximation vol. 16*, 3 (2000), 359–397.