# Piecewise smooth subdivision surfaces with normal control

Henning Biermann
NYU

Adi Levin
Tel-Aviv University

Denis Zorin
NYU

## Abstract

In this paper we introduce improved rules for Catmull-Clark and Loop subdivision that overcome several problems with the original schemes (lack of smoothness at extraordinary boundary vertices, folds near concave corners). In addition, our approach to rule modification allows generation of surfaces with prescribed normals, both on the boundary and in the interior, which considerably improves control of the shape of surfaces.

## 1 Introduction

Subdivision surfaces have rapidly gained popularity in computer graphics. One of the main advantages of subdivision algorithms is that they are capable of efficiently generating smooth surfaces from an arbitrary initial mesh. Subdivision algorithms are also attractive because they are conceptually simple and can be easily modified to create surface features without making major changes to the algorithm.

A general surface representation should behave in a predictable way for arbitrary initial data. For example, the properties of spline surfaces are well understood for arbitrary regular grids of control points. The variety of data that can serve as the input for most common subdivision algorithms is considerably larger: an arbitrary polygonal or triangular mesh with a boundary, possibly with marked edges and vertices. While the general structure of the algorithm does not depend much on the initial data, the rules should account for the different local mesh configurations.

Subdivision rules for the interior part of a control mesh are well understood, while the boundary rules have received relatively little attention. Boundary rules are quite important for a variety of reasons. The boundary of the surface, together with the contour lines, forms the visual outline. While the interior of the surface often has to be defined only approximately, the boundary conditions may be significantly more restrictive; for example, it is often necessary to join several surfaces along their boundaries. Boundary subdivision rules immediately lead to the rules for sharp creases. Using the techniques described in [3], one can also generate soft creases using essentially the same rules.

In addition to specifying the boundary or crease curves, it is often desirable to specify the tangent planes on the boundary. Existing subdivision schemes do not allow one to control the tangent plane behavior.

The goal of this paper is to present two complete sets of subdivision rules for generating piecewise-smooth, $C^1$-continuous, almost everywhere $C^2$ subdivision surfaces, with tangent plane control. Our rules extend the well-known subdivision Catmull-Clark[2] and Loop [8] subdivision schemes. A complete $C^1$-continuity analysis of our schemes will be presented elsewhere [17]. Our rules allow modeling surfaces with piecewise-smooth boundaries and different types of corner vertices and prescribed normals on on the boundary or the interior. At the same time, only minimal changes are introduced to the basic Catmull-Clark and Loop algorithms (with crease modifications of [7] and [3]).

We use a uniform approach to derive the desired subdivision rules, which can be applied to any stationary subdivision scheme. In this paper, we focus on the Loop and Catmull-Clark subdivision schemes as schemes having the greatest practical importance.

## 2 Previous Work

A number of subdivision schemes have been proposed over years since Catmull and Clark introduced subdivision surfaces in 1978 ([2]). We refer the reader to a more detailed survey of subdivision in [1].

Theoretical analysis of subdivision rules was performed in [13, 12, 6, 14, 16]. Most theoretical work has focused on subdivision of closed surfaces; almost all theoretical analysis relies on the rotational symmetry of the subdivision rules and applies only to the interior rules. A noteable exception is the work of Schweitzer [14] and the recent work of Levin.

Subdivision rules for Doo-Sabin dual subdivision surfaces for the boundary were discussed by Doo [4] and Nasri [9, 10], but only partial theoretical analysis was performed.

Our work builds on the work of Hoppe et al. [7] and partially on the ideas of Nasri [11].

To the best of our knowledge, the subdivision rules proposed in work [7] are the only ones that result in provably $C^1$-continuous surfaces (the analysis can be found in Schweitzer [14]). However, these rules suffer from two problems:

—the shape of the boundary of the generated surface depends on the control points in the interior;

—only one rule for corners is defined, which works well for convex corners but does not work well for concave corners (see Section 4).

Standard Catmull-Clark rules, when applied to the boundary, suffer from the same problems.

Recently, a generalization of Catmull-Clark and Doo-Sabin subdivision rules that contains NURBS as a subset was proposed by Sederberg et al. [15]. While including NURBS patches as a special case is important for applications that require use of existing NURBS models, it comes at a price: the complexity of the algorithms is increased and the behavior of the surface near the extraordinary points becomes difficult to analyze and predict. The smooth crease effects that are obtained by manipulating NURBS weights for subdivision surfaces can be achieved without introducing nonuniform knot spacing using an elegant technique proposed by DeRose et al. [3].

## 3 Piecewise-smooth surfaces and subdivision

**Piecewise smooth surfaces.** In order to derive subdivision rules we need to describe more precisely the type of the surfaces that we wish to model. We have chosen a class of *piecewise smooth surfaces*. This is the class of surfaces that includes, for example, quadrilateral free-form patches, and other common modeling primitives. At the same time, we have excluded from consideration surfaces with various types of singularities.

Here we give an informal description of piecewise-smooth surfaces; more mathematical details can be found in [17]. To define this class, we start with smooth surfaces that have a piecewise-smooth boundary. For simplicity, assume that our surfaces do not have self-intersections. Recall that for closed $C^1$-continuous surface $M$ in $\mathbf{R}^3$ each point has a neighborhood that can be smoothly deformed into an open planar disk $D$.

*A surface with a smooth boundary* is defined in a similar way, but the neighborhoods of points on the boundary can be smoothly deformed into a half-disk $H$, with closed boundary. To define a

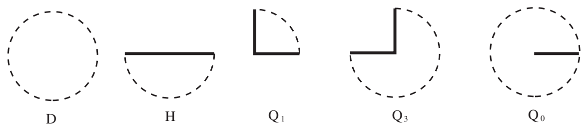


Figure 1: The charts for a surface with piecewise smooth boundary.

surface with piecewise smooth boundaries, we introduce two additional types of local charts: concave and convex corner charts, $Q_3$ and $Q_1$ (Figure 1). Thus, a $C^1$-continuous surface with piecewise smooth boundary locally looks like one of the domains $D$, $H$, $Q_1$ and $Q_3$.

*Piecewise-smooth surfaces* are the surfaces that can be constructed out of surfaces with piecewise smooth boundaries joined together.

If the resulting surface is not $C^1$-continuous at the common boundary of two pieces, this common boundary is a crease. We allow two adjacent smooth segments of a boundary to be joined, producing a crease ending in a *dart* (cf. [7]). For dart vertices an additional chart $Q_0$ is required; the surface near a dart can be deformed into this chart smoothly everywhere except at an open edge starting at the center of the disk.

An important observation for constructing subdivision rules for the boundary is that the last two corner types are not equivalent, that is, there is no smooth *nondegenerate* map from $Q_1$ to $Q_3$. It follows from the theory of subdivision [18], that a single subdivision rule cannot produce both types of corners. In general, any complete set of subdivision rules should contain separate rules for all chart types. Most, if not all, known schemes miss some of the necessary rules.

# 4 Problems with standard rules

In this section we demonstrate the problems that occur when some common rules are used; it turns out that not all piecewise-smooth surfaces can be adequately represented.

**Concave corners.** Suppose we wish our surface to have concave corner at a vertex of the boundary; this often happens when one would like to make a surface with a hole. In this case, it is natural to arrange the control points into a concave configuration, and expect the generated surface to approximate this configuration. However, if the simplest type of corner rules is used, such as the rules described in [7], it turns out that these rules can generate only convex corners. If the initial mesh has a concave corner, the rules force the surface to approach the corner from the outer, convex, side. The surface develops a fold (Figure 2).

**Boundary rules.** It was observed by Hoppe et al.[7] that the surface is not smooth at the extraordinary vertices if the standard rules are used. They propose a simple solution to this problem – a modification of the coefficients used to compute the boundary curve. An undesirable consequence of this fact is that the boundary curve is no longer independent from the interior of the surface. After the modification, the boundary curve depends on the number of the interior vertices adjacent to each boundary vertex. If two separate meshes have identical boundaries, a gap between the meshes can appear as a result of subdivision (Figure 10a).

If the original rules are retained, the surface does not have a well-defined tangent plane at the point. This becomes apparent when the boundary at a point is twisted as shown in Figure 10b. However, even if the smooth rules of [7] are used, the twist persists, although in the limit the surface is formally smooth. The reason for this can be understood by examining the eigenstructure of the subdivision matrix:

## 4.1 Subdivision matrix

Our method for constructing subdivision rules is based on manipulating the eigenstructure of the *subdivision matrix*; this idea can be traced back to [5]. Consider a vertex $v$ of degree $k$, and let $p$ be the vector of control points in a neighborhood of the vertex (Figure 4).

Recall that the subdivision matrix $S$ is the matrix of subdivision coefficients relating the vector of control points $p^j$ to the vector of control points $p^{j+1}$ on a similar neighborhood on the next subdivision level. Suppose the size of the matrix is $N$. Properties of subdivision are closely related to the structure of the matrix. This is easy to see if we decompose the vector of control points $p$ with respect to the eigenbasis $\{x^m\}$, $m = 0..N-1$, of $S$, if one exists:

$$p = \mathbf{a}_0 x^0 + \mathbf{a}_1 x^1 + \mathbf{a}_2 x^2 + \ldots \qquad (4.1)$$

Note that the coefficients $\mathbf{a}_i$ in this decomposition are 3d vectors, because $p$ is a vector of 3d control points. We assume that the eigenvectors $x^j$ are arranged in the order of nonincreasing eigenvalues, and the first eigenvalue $\lambda_0$ is 1, which is required for convergence of subdivision. Subdividing the surface $l$ times means that the subdivision matrix is applied to $p$ $l$ times:

$$S^l p = \lambda_0^l \mathbf{a}_0 x^0 + \lambda_1^l \mathbf{a}_1 x^1 + \lambda_2^l \mathbf{a}_2 x^2 + \cdots \qquad (4.2)$$

If we further assume that $\lambda_1 = \lambda_2 = \lambda$ and $\lambda > |\lambda_3|$, it can be immediately seen from this formula that the vector of control points $p^l$ in the neighborhood of the vertex $v$ can be approximated by $\mathbf{a}_0 x^0 + \lambda^l(\mathbf{a}_1 x^1 + \mathbf{a}_2)$. If $\mathbf{a}_1 \times \mathbf{a}_2$ is not zero, all the control points $p_i^j$ are close to the plane passing through $\mathbf{a}_0$ and spanned by vectors $\mathbf{a}_1$ and $\mathbf{a}_2$. As $l \to \infty$ the positions of all points converge to $a_0$. This means that to compute the limit position for a vertex, we simply need to compute $\mathbf{a}_0$ and to compute the tangent vectors, we need to compute $\mathbf{a}_1$ and $\mathbf{a}_2$. This can be achieved using the left eigenvectors of $S$, i.e. vectors $l^i$, satisfying $(l^i, x^i) = 1$ and $(l^i, x^j) = 0$ if $i \neq j$.

From equation 4.2 we can see that the tangent plane of the surface can be influenced by changing eigenvectors $x^1$, $x^2$ or $l^1$, $l^2$, eigenvalue $\lambda$, or altering $\mathbf{a}_1$ and $\mathbf{a}_2$ in some other way.

# 5 Constructing subdivision rules

Both for obtaining the rules for corners and modification of normals we choose a way in which we would like to modify the coefficients $\mathbf{a}_1$ and $\mathbf{a}_2$.

**Smooth boundary points.** If we use standard Loop subdivision rules near the boundary, the resulting surface is not tangent plane continuous. The reason for this is that the tangent vector of the boundary curve corresponds to the eigenvalue $1/2$ of the subdivision matrix which is not subdominant. To make the surface $C^1$-continuous, we can modify the eigenvalues in such a way that $1/2$ becomes subdominant. This can be achieved by scaling all coefficients $\mathbf{a}_i$ corresponding to eigenvalues different from 1, $1/2$ and the first subdominant value by a sufficiently small factor. There are many ways to accomplish this; we choose a way that leads to the simplest subdivision rules.

**Corners.** For both concave and convex corners the tangent vectors to the boundary curves meeting at the corner do not coincide. If the surface is smooth, this means that the tangent plane to the surface is completely defined by the boundary curves. Therefore, the eigenvalues corresponding to the two tangent vectors of the boundary have to be subdominant. If we use standard rules for endpoint-interpolating spline on the boundary, we just have to ensure that $1/2$ is the subdominant eigenvalue, because both both eigenvectors defining rules for computing the tangents on the boundary have eigenvalue $1/2$.

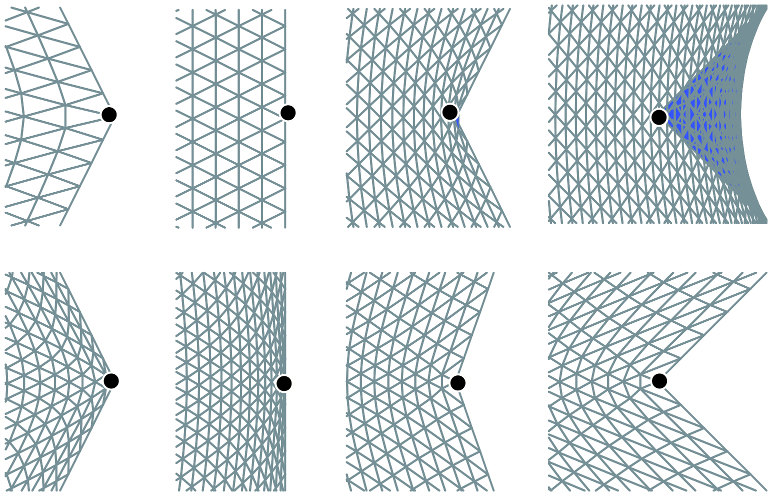

Figure 2: Upper row: behavior of a subdivision surface when rules of [7] are applied near a corner of the control mesh. As the corner of the control mesh is moved, the surface develops a fold. Lower row: our concave corner rules rules applied to the same mesh have the inverse behavior; when the corner is convex, a small fold develops (not visible in the picture), and the mesh has the most natural configuration for a concave corner.

**Normal modification.** If a normal $n$ is prescribed at a vertex (internal or crease), we need to modify $\mathbf{a}_1$ and $\mathbf{a}_2$ in such a way that the cross product $\mathbf{a}_1 \times \mathbf{a}_2$ points along the normal. This can be achieved if we replace $\mathbf{a}_i$ $i = 1,2$ with $\mathbf{a}_i - n(\mathbf{a}_i, n)$, eliminating the components of the tangent vectors along the normal. Using the fact that $\mathbf{a}_i = (l_i, p)$, we can see that this modification can be regarded as a special type of subdivision rules. [1]

# 6 Subdivision rules

In this section we describe the subdivision rules that we have derived following the ideas of the previous section. More details can be found in [17].

## 6.1 Tagged meshes

We use a *tagged mesh* to define our rules. There are three types of tags, one associated with edges, another with vertices and the last with edge-vertex pairs.

- *edge tags* indicate that the subdivision surface should have a crease at this edge; we tag all the boundary edges as creases.
- *vertex tags* label vertices as smooth crease/boundary or corner.
- *edge-vertex tags* gives the type of the corner (convex or concave) the surface should have as it approaches the vertex from the direction of the edge; edge-vertex tags can be assigned only to edges without edge tags.

The edge-vertex tags are used only for edges that have a single endpoint on a crease. If we consider a vertex with at least one adjacent tagged edge, all incident edges can be separated into sectors. The edge-vertex tags in each sector (Figure 5) are identical.

Note the difference between our system of tagging and the one presented in [7]: because we have different corner types it is necessary to tag edge-vertex pairs, rather than mesh vertices themselves.

[1]The rules derived from these conditions are no longer applied to each coordinate of the control points separately: rather, the whole 3d vector is required. The coefficients of subdivision in this case are matrices rather than scalars

For triangular meshes (Loop subdivision) we define two types of rules: *edge rules* that are applied to compute positions of new control points added at each subdivision step, and *vertex rules* that are applied to compute control points for vertices already present in the mesh. There are three vertex rules: interior smooth, crease, and corner; there are five edge rules: smooth, crease, crease neighbor, convex corner neighbor and concave corner neighbor. The last four types apply to vertices that are inserted next to a crease vertex, but are not crease vertices themselves (Figure 6).

For quadrilateral meshes (Catmull-Clark subdivision) in addition to the edge and vertex rules we need to use one more rule to insert control points into faces.

For an arbitrary polygonal mesh, we assume that either face triangulation or standard Catmull-Clark rules are used to convert it to a triangular or a quadrilateral mesh respectively. This solution is not entirely satisfactory. Further investigation is required to choose the best top-level rules.

**Choosing subdivision rules.** To compute face points on quadrilateral meshes, we always use the standard Catmull-Clark rules. The procedure for choosing the edge and vertex subdivision rules is identical for both schemes.

To determine the vertex rule it is sufficient to look at the vertex tag. If the vertex is tagged as boundary or crease, the standard spline rule is applied; if it is tagged as a corner vertex, then we use the interpolating rule: the control point on subdivision level $j$ coincides with the control point on subdivision level $j-1$.

To determine the edge rule for a newly inserted vertex, we need to examine the edge on which it was inserted. For a crease edge, we always use the crease subdivision rule. Otherwise, we need to know the tags of the surrounding vertices.

Let $v_1$ and $v_2$ be the endpoints of the edge $e$.

- If no edge-vertex tags are present, and both vertices are interior, we use the smooth edge rule.
- If no tags are present, and one of the vertices $v_1$, $v_2$ is a boundary vertex, we use the crease neighbor rule.
- If a single vertex has a vertex-edge tag, we use convex or concave corner neighbor rule, depending on the tag.

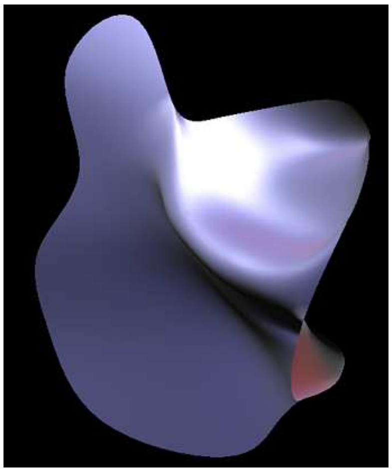

Figure 3: A surface with a twist (in the lower left part of the image)

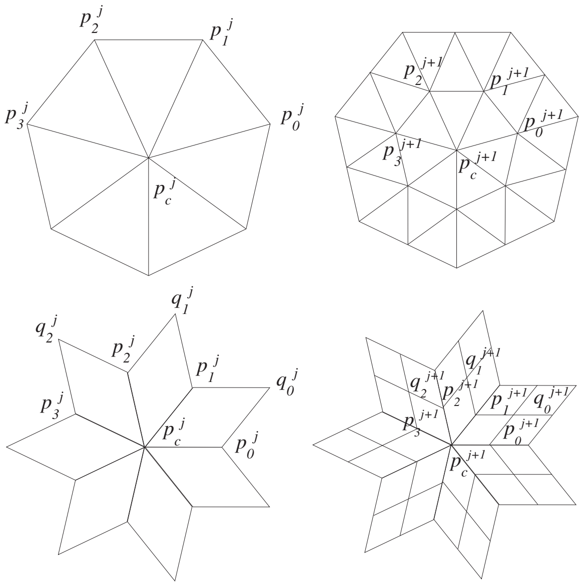


Figure 4: Similar neighborhoods of a vertex of degree $k$ on different subdivision levels. The subdivision matrix $S$ relates the vector of control points $p^j = \left[p_c^j, p_0^j, \dots p_k^j\right]$ to the vector of control points on level $j+1$ $p^{j+1}$.

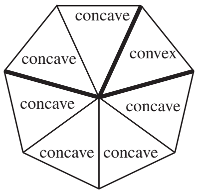


Figure 5: Edge-vertex tags. If several tagged edges meet at a vertex, all non-crease edges in each sector between tagged edges receive either convex or concave tag. Note that the tag does not necessarily correspond to the convexity or concavity of the configuration of vertices. Concave tags on sectors of size 1 are not allowed.

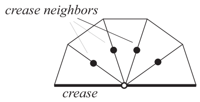


Figure 6: We call vertices inserted on the edges with a single endpoint on the crease *crease neighbors*. If the endpoint is a corner vertex, corresponding neighbors are called convex or concave corner neighbors, depending on the type of the sector in which they are located.

- If both vertices have tags, or are on the boundary, we use the average of the rules determined for each vertex.

Now we describe vertex and edge rules. In the definition of the crease neighbor edge rules we use the *crease degree of the vertex* which we define as the number of triangles in the sector that we are considering.

## 6.2 Vertex rules

As we have mentioned above, there are three vertex rules: interior, crease and corner. The interior rules for both schemes coincide with the standard Catmull-Clark and Loop rules respectively Figure 7. For both Loop and Catmull-Clark schemes we use the standard cubic B-spline subdivision rules as crease rules (Figure 9).

Interior smooth rule is also applied to the dart vertices. Thus, only smooth darts are allowed.

## 6.3 Edge Rules

The crease rules are the standard cubic B-spline subdivision rules.

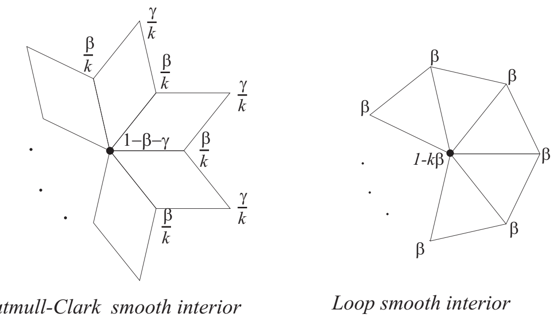


Figure 7: Smooth interior vertex rules for the triangular and quadrilateral schemes; these are the standard Loop and Catmull-Clark subdivision rules. For the Loop scheme, $\beta$ can be chosen to be $3/8k$, where $k$ is the degree, for $k > 3$. For $k = 3$, $\beta$ can be taken to be $3/16$. For the Catmull-Clark scheme, $\beta = \frac{3}{2k}$ and $\gamma(\theta) = \frac{1}{4k}$.

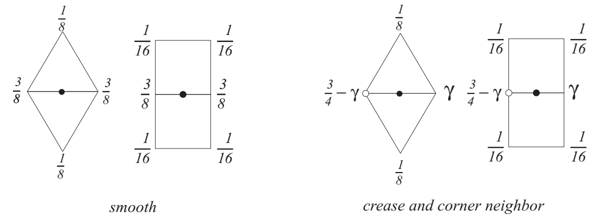


Figure 8: Edge rules for the triangular and quadrilateral schemes (before flatness and normal modification); smooth rules are the standard Loop and Catmull-Clark subdivision rules. Rules for crease and corner neighbors use different $\gamma$ depending on the rule type.

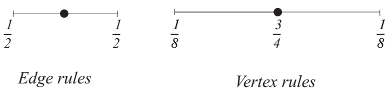


Figure 9: Subdivision rules for smooth crease vertices for both triangular and quadrilateral schemes. These rules always coincide with the cubic B-spline subdivision. Corner vertices are interpolated.

We choose crease neighbor, concave corner neighbor and convex corner neighbor rules in a uniform way. While the rules can be applied in a single step, it is convenient to separate them into two parts: initial calculation using the Loop or Catmull-Clark stencil, and subsequent modification to ensure $C^1$-continuity or improve convergence to the tangent plane.

The stencil that we use on the first step is almost identical to the standard rules; the only difference is a new parameter $\gamma(\theta)$ (Figure 8). For the Loop scheme, $\gamma(\theta) = 1/2 - 1/4\cos\theta$; For the Catmull-Clark scheme, we use $\gamma = 3/8 - 1/4\cos\theta$. For a vertex of crease degree $k$, we have chosen the following values of $\theta$, which result in the best-shaped limit configurations of control points (see [17] for details):

- $\theta = \alpha/k$ for convex corners where $\alpha$ is the angle between the crease edges meeting at the corner.
- $\theta = \pi/k = \theta_k$ for smooth crease vertices.
- $\theta = (\pi-\alpha)/k$ for concave corner vertices.

As the second step, we modify the positions of control points using the following general rule parameterized by a parameter $s$, (the *flatness parameter*):

$$\left(p^{j+1}\right)_{\text{new}} = (1-s)\left(p^{j+1}\right)_{\text{old}} + s\sum_{m=0}^{k}(l_m^0x^0 + l_m^1x^1 + l_m^2x^2)p_m^{j+1}$$

(Loop)

$$\left(p^{j+1}\right)_{\text{new}} = (1-s)\left(p^{j+1}\right)_{\text{old}} + s\sum_{m=0}^{k}(l_m^{0p}x^0 + l_m^{1p}x^1 + l_m^{2p}x^2)p_m^{j+1}$$
$$+ s\sum_{m=0}^{k-1}(l_m^{0q}x^0 + l_m^{1q}x^1 + l_m^{2q}x^2)q_m^{j+1} \qquad \text{(Catmull-Clark)}$$

(6.1)

where $l^0$,$l^1$, $l^2$, $x^0$, $x^1$, $x^2$ are the eigenvectors given in Appendix A. As a result of this modification, all eigenvalues of the subdivision matrix are scaled by the factor $1-s$, excluding In the extreme case, when $s = 1$, the control points are immediately projected to the tangent plane. The flatness parameter can be used either as a modeling parameter, or chosen to be $1 - 1/4(\gamma + 1/4\cos 2\theta_k)$ for both schemes this value is optimal in a sense that the resulting scheme has bounded curvature variation; for larger values of $s$ the curvature is unbounded, for smaller values it is 0.

For convex and smooth boundary rules, the value of $s$ can be arbitrary between 0 and 1. To ensure $C^1$ continuity,a constraint has to be imposed on the parameter $s$:

**Crease and convex corner neighbor rules.** If we use $s = 0$ for the smooth crease vertices, which results in the simplest subdivision rules (the modification step is absent), then formally $C^1$-continuous surfaces are visually indistinguishable from non-$C^1$-continuous, obtained by applying standard Loop or Catmull-Clark edge rules. This was previously observed in [3]. The absence of tangent plane continuity can only be seen for meshes with a twist on the boundary (Figure 3), and many subdivision steps are required before the difference between the non-smooth and smooth rules becomes apparent. However, in applications in which numerical, rather than just visual, continuity of the normal is desired, it is still preferable to use the smooth rules, even with $s = 0$. For example, we have observed that in contour line calculations better results are achieved if the smooth rules are used. Finally, if a smoother transition is desired at the twist points, parameter $s$ can be used to increase the speed at which the surface converges to the tangent plane.

**Concave corner neighbor rules.** Recall that these rules are applied to insert control points on nontagged edges adjacent to a concave corner vertex. This is the only case when using non-zero $s$ is required for obtaining $C^1$-continuous surfaces; the condition on $s$ is $(1-s)(2+\cos\theta_k - \cos\theta) < 2$.

### 6.4 Normal control

As it was explained in section 5, the subdivision rules can be adjusted to produce surfaces with prescribed normals. Suppose a vertex $v$ of degree $k$ (or crease degree $k$, if it is a crease vertex) has a prescribed normal. The subdivision rules producing the desired effect can be best defined in two steps, in a way similar to the rules for crease neighbor vertices. For an interior vertex $v$, the only control points whose positions need to be modified are the control points $p_i^j$, $i = 0, \ldots k-1$, immediately adjacent to $v$. The modification is performed using the following rules:

$$\left(p^{j+1}\right)_{\text{new}} = \left(p^{j+1}\right)_{\text{old}} - s\mathbf{n}\sum_{m=0}^{k}(l_m^0x^0 + l_m^1x^1 + l_m^2x^2)(p_m^{j+1}, \mathbf{n})$$

(Loop)

$$\left(p^{j+1}\right)_{\text{new}} = \left(p^{j+1}\right)_{\text{old}} - s\mathbf{n}\sum_{m=0}^{k}((l_m^{0p}x^0 + l_m^{1p}x^1 + l_m^{2p}x^2)p_m^{j+1}, \mathbf{n})$$
$$+ s\mathbf{n}\sum_{m=0}^{k-1}((l_m^{0q}x^0 + l_m^{1q}x^1 + l_m^{2q}x^2)(q_m^{j+1}, \mathbf{n}) \qquad \text{(Catmull-Clark)}$$

(6.2)

This modification is similar to the modifications described in the previous section. The parameter $s$ determines how fast the normal converges to the prescribed limit position. If $s = 1$, after one step of subdivision the normal computed using standard coefficients $l_m^1$, $l_m^2$ coincides with the prescribed normal.

One can observe that the modification moves the points towards a plane perpendicular to the normal. Our analysis shows that the limit position for the vertex $v$ remains unchanged, if no normal is prescribed for adjacent vertices. If the normals are modified on the boundary, the boundary curves remain B-splines, but with different control points.

If the normals are prescribed for two endpoints of an edge $e$, on the first subdivision step we take the average of the two modifications for edge vertex inserted on $e$. On subsequent steps no two vertices with prescribed normals share an edge.

# 7 Results

A complex corner vertex with two different types of corners meeting is shown in Figure 12a. Figure 12b shows a plane with an interior and boundary normals modified. The results of prescribing the normals on the boundary of a surface are shown in Figure 13 and Figure 14 for the Loop and Catmull-Clark surfaces.

# 8 Conclusion and Future Work

We have presented a simple modification of the two most popular subdivision schemes that improves the behavior of the generated surfaces on the boundary and creases and provides additional controls for surface modeling.

While the class of surfaces that was considered in the paper is quite general and sufficient for most purposes, we have excluded many types of surface singularities which might be useful for modeling purposes. It would be useful to explore which singularities are useful for modeling purpose and construct subdivision rules to create such singularities.

# References


[1] Subdivision for computer graphics and animation. SIGGRAPH '98 Course Notes.

[2] Ed Catmull and James Clark. Recursively generated B-spline surfaces on arbitrary topological meshes. *Computer Aided Design*, 10(6):350–355, 1978.

[3] Tony DeRose, Michael Kass, and Tien Truong. Subdivision surfaces in character animation. In Michael Cohen, editor, *SIGGRAPH 98 Conference Proceedings*, Annual Conference Series, pages 85–94. ACM SIGGRAPH, Addison Wesley, July 1998. ISBN 0-89791-999-8.

[4] D. Doo. A subdivision algorithm for smoothing down irregularly shaped polyhedrons. In *Proceedings on Interactive Techniques in Computer Aided Design*, pages 157–165, Bologna, 1978.

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

[6] Ayman Habib and Joe Warren. Edge and vertex insertion for a class of subdivision surfaces. Preprint. Computer Science, Rice University, 1996.

[7] Hugues Hoppe, Tony DeRose, Tom Duchamp, Mark Halstead, Huber Jin, John McDonald, Jean Schweitzer, and Werner Stuetzle. Piecewise smooth surface reconsruction. In *Computer Graphics Proceedings*, Annual Conference Series, pages 295–302. ACM Siggraph, 1994.

[8] Charles Loop. Smooth subdivision surfaces based on triangles. Master's thesis, University of Utah, Department of Mathematics, 1987.

[9] Ahmad H. Nasri. Polyhedral subdivision methods for freeform surfaces. *ACM Transactions on Graphics*, 6(1):29–73, January 1987.

[10] Ahmad H. Nasri. Boundary corner control in recursive subdivision surfaces. *Computer Aided Design*, 23(6):405–410, 1991.

[11] Ahmad H. Nasri. Surface interpolation on irregular networks with normal conditions. *Computer Aided Geometric Design*, 8:89–96, 1991.

[12] Jörg Peters and Ulrich Reif. Analysis of generalized B-spline subdivision algorithms. *SIAM Jornal of Numerical Analysis*, 1997.

[13] Ulrich Reif. A unified approach to subdivision algorithms near extraordinary points. *Computer Aided Geometric Design*, 12:153–174, 1995.

[14] J. E. Schweitzer. *Analysis and Application of Subdivision Surfaces*. PhD thesis, University of Washington, Seattle, 1996.

[15] Thomas W. Sederberg, Jianmin Zheng, David Sewell, and Malcolm Sabin. Non-uniform recursive subdivision surfaces. In Michael Cohen, editor, *SIGGRAPH 98 Conference Proceedings*, Annual Conference Series, pages 387–394. ACM SIGGRAPH, Addison Wesley, July 1998. ISBN 0-89791-999-8.

[16] Denis Zorin. Smoothness of subdivision on irregular meshes. *Constructive Approximation*, 1998. submitted.

[17] Denis Zorin and Tom Duchamp. Smoothness of subdivision surfaces on the boundary. in preparation, 1999.

[18] Denis N. Zorin. *Subdivision and Multiresolution Surface Representations*. PhD thesis, Caltech, Pasadena, California, 1997.


# A Coefficients for left and right subdominant eigenvectors

Here we list the coefficients required to compute the rules described in the paper. Recall that the subscript $c$ denotes the centeral vertex of a ring of vertices, and $p_i$ are the control points in that ring.

For both schemes and all rules the right eigenvector $x^1$ has all entries equal to 1.

**Loop scheme.**

- Interior vertex of degree $k$. In all cases $i = 0..k-1$:

$$l_c^0 = \frac{1}{1+(8k/3)\beta}\; l_i^0 = \frac{(8/3)\beta}{1+(8k/3)\beta}$$

$$x_c^1 = x_c^2 = l_c^1 = l_c^2 = 0$$

$$x_i^1 = \cos 2i\theta_k, x_i^2 = \sin 2i\theta_k$$

$$l_i^1 = \frac{2}{k}\cos 2i\theta_k, l_i^2 = \frac{2}{k}\sin 2i\theta_k$$

- Smooth crease vertex of crease degree $k$. In this and the next case case $i$ varies from 0 to $k$.

$$l_c^0 = 2/3,\, l_1^0 = l_k^0 = 1/6,\, l_i^0 = 0, \text{for } i = 1\dots k-1$$

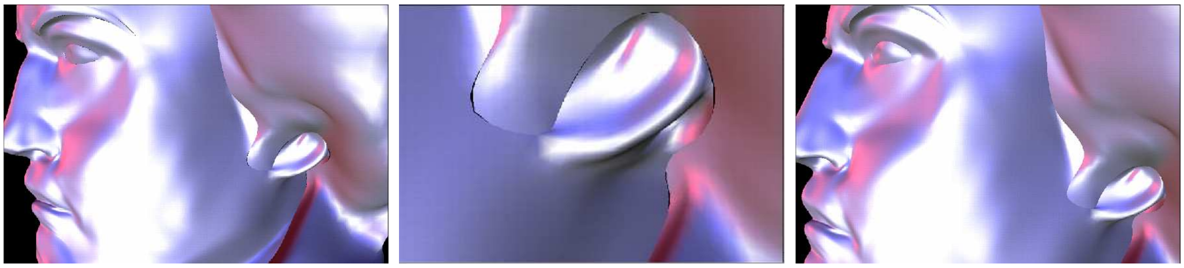

Figure 10: Behavior of subdivision rules near extraordinary points on the boundary. a-b. Application of the rules of [7] creates a gap in the mesh. With our rules no gap is created. c. Detail of the mesh: both rules of [7] and unmodified rules result in a twist in the surface.

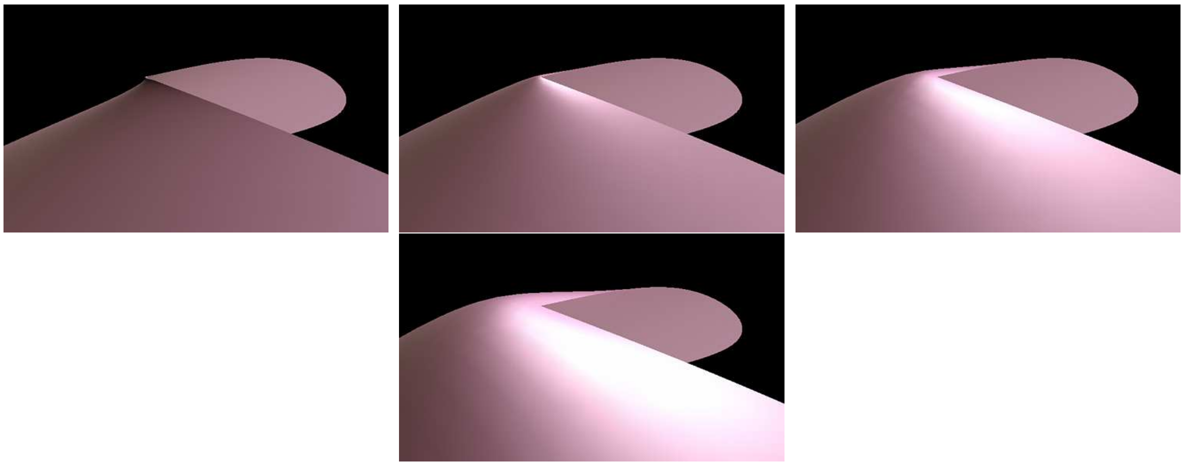

Figure 11: This figure shows the result of applying our rules to a generate a concave corner. Results for several values of the flatness parameter are shown. The first image corresponds to $s = 0$. i.e. to a scheme almost identical to [7].

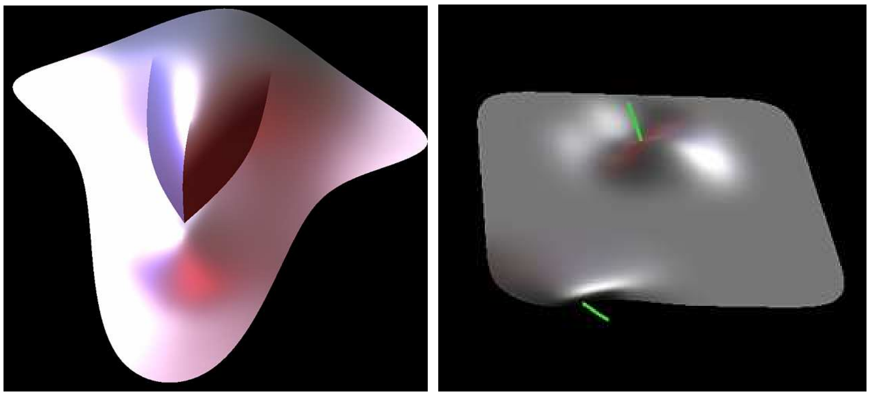

Figure 12: a: An interior corner vertex with two convex corners and a concave corner meeting. b: modification of normals to a plane at an interior and a boundary point.

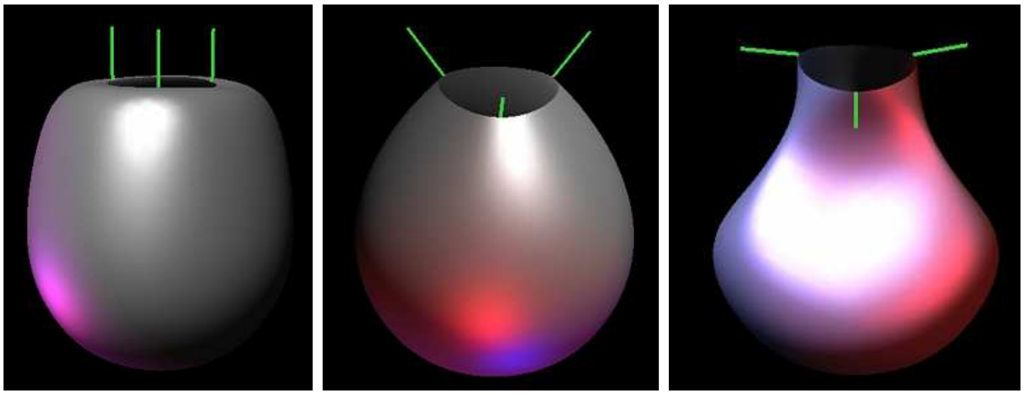

Figure 13: Modification of normals for the Loop scheme; the original mesh is an icosahedron. Left: all normals pointing up. Middle: Normals horizontal. Bottom: all normals pointing down.

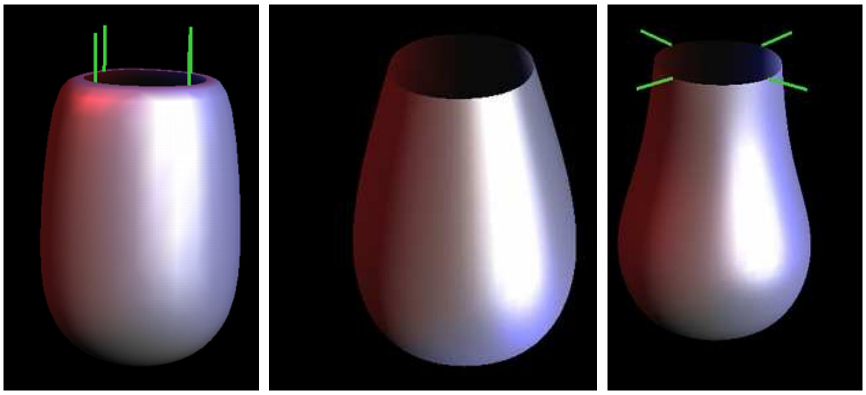

Figure 14: Modification of normals for the Catmull-Clark scheme; Left: all normals pointing up. Middle: no normal modification. Right: all normals horizontal.

For $k=1$,

$$x^1=[\frac{2}{3},-\frac{1}{3},\frac{2}{3}],\quad x^2=[1,0,-1]$$

$$l^1=[\frac{1}{2},-1,\frac{1}{2}],\quad l^2=[\frac{1}{2},0,-\frac{1}{2}],$$

otherwise $x_c^1=x_c^2=l_c^1=0$,

$$x_i^1=\cos i\theta_k, x_i^2=\sin i\theta_k$$

$$l_0^1=1/2,\ l_k^1=-1/2,\quad l_i^1=0$$

$$l_0^2=l_k^2=-\frac{2}{k}\left(\left(\frac{a}{2}+\frac{1}{6}\right)\sigma_1+\frac{1}{2}b\sigma_3\right)$$

$$l_c^2=-\frac{2}{k}\left(\left(\frac{2}{3}-a\right)\sigma_1-b\sigma_3\right)$$

$$l_i^2=\frac{2}{k}\sin i\theta_k$$

, where

$$a=\frac{\frac{1}{4}(1+\cos\theta_k)}{3(\frac{1}{2}-\frac{1}{4}\cos\theta_k)}\qquad b=\frac{\frac{2}{3}-a}{cos\frac{k\zeta}{2}}$$

$$\sigma_1=\frac{\sin\theta_k}{1-\cos\theta_k}$$

$$\sigma_3=\frac{\cos\frac{k\zeta}{2}\sin\theta_k}{\cos\zeta-\cos\theta_k}$$

$$\zeta=\arccos(\cos\theta_k-1)$$

- Convex/concave corner vertex of crease degree $k$ and with rule parameter $\theta\neq\theta_k$:

$$l_c^0=1,\ l_i^0=0, k=0\ldots k$$

$$x_c^1=x_c^2=0,\quad x_i^1=\frac{\sin i\theta}{\sin k\theta}, x_i^2=\frac{\sin(k-i)\theta}{\sin k\theta},\quad i=0\ldots k$$

$$l_0^1=0,\ l_k^1=1\ l_c^1=-1\quad l_i^1=0,\ i=1\ldots k-1$$

$$l_0^2=1,\ l_k^2=0\ l_c^2=-1\quad l_i^2=0,\ i=1\ldots k-1$$

**Catmull-Clark scheme**

- Interior vertex of degree $k$. $i$ varies from 0 to $k-1$.

$$l_c^0=\frac{k}{k+5},\ l_{pi}^0=\frac{4}{k(k+5)},\ l_{qi}^0=\frac{1}{k(k+5)}$$

$$x_c^1=x_c^2=l_c^1=l_c^2=0$$

$$x_{pi}^1=\cos 2i\theta_k, x_{pi}^2=\sin 2i\theta_k$$

,

$$x_{qi}^1=\frac{1}{4\lambda-1}(\cos 2i\theta_k+\cos 2(i+1)\theta_k)$$

$$x_{qi}^2=\sin 2i\theta_k\quad\frac{1}{4\lambda-1}(\cos 2i\theta_k+\cos 2(i+1)\theta_k)$$

where $\lambda=5/16+1/16\cos 2\theta_k+\cos\theta_k\sqrt{2(9+\cos 2\theta_k)}$.

- Smooth crease vertex of crease degree $k$; in this and the next case, $i$ varies from 0 to $k$ for edge components $x_{p\,j}^1$ and from 0 to $k-1$ for face components.

If $k>1$,

$$x_c^1=0,\quad x_{pi}^1=\sin i\theta_k,\quad x_{q,i}^1=\sin i\theta_k+\sin(i+1)\theta_k$$

, if $k=1$ $x^1=[1/18,-1/9,-1/9,-5/18]$.

For $k>1$,

$$l_0^1=l_k^1=-R(1+2\cos\theta_k)\quad l_c^1=4R(-1+\cos\theta_k)$$

$$l_{pi}^1=\frac{4\sin i\theta_k}{(3+\cos\theta_k)k}$$

$$l_{fi}^1=\frac{\sin i\theta_k+\sin(i+1)\theta_k}{(3+\cos\theta_k)k}$$

where $R=\frac{\cos\theta_k+1}{\sin\theta_k(3+\cos\theta_k)k}$

For $k=1$, $l^1=[6,-3,-3,0]$

If $k>1$,

$$x_c^2=0,\quad x_{pi}^2=\cos i\theta_k,\quad x_{q,i}^1=\cos i\theta_k+\cos(i+1)\theta_k$$

, if $k=1$ $x^1=[0,-1/2,1/2,0]$.

$$l_c^2=0, l_{p0}^2=1/2, l_{pk}^2=-1/2$$

for $k=1$, $l^2=[0,-1,1,0]$

- Convex/concave corner vertex of crease degree $k$ and with rule parameter $\theta\neq\theta_k$

  Left eigenvectors are the same as for Loop, with zeroes everywhere except $l_c$, $l_0$ and $l_k$. The right eigenvectors are

$$x_c^1=x_c^2=0,\quad x_{pi}^1=\frac{\sin i\theta}{\sin k\theta}, x_{qi}^1=\frac{(4\lambda-1)\sin k\theta}{(\sin i\theta+\sin(i+1)\theta)}$$

$$x_{pi}^1=\sin(k-i)\theta\sin k\theta, x_{qi}^1=\frac{(4\lambda-1)\sin k\theta}{(\sin(k-i)\theta+\sin(k-i-1)\theta)}$$