Piecewise Smooth Subdivision Surfaces with Normal Control

Henning Biermann*
New York University

Adi Levin†
Tel Aviv University

Denis Zorin‡
New York University

Abstract

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

CR Categories and Subject Descriptors: I.3.5 [**Computer Graphics**]: Computational Geometry and Object Modeling – Curve, surface, solid, and object representations; Boundary representations.

Additional Keywords: Subdivision surfaces, boundary control.

1 Introduction

Subdivision surfaces are rapidly gaining popularity in computer graphics. A number of commercial systems use subdivision as a surface representation: Alias|Wavefront's Maya, Pixar's Renderman, Nichimen's Mirai, and Micropace' Lightwave 3D, to name just a few. The greatest advantage of subdivision algorithms is that they efficiently generate smooth surfaces from arbitrary initial meshes. 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.

At the same time, one of the drawbacks of subdivision is a lack of precise definition of the schemes with guaranteed behavior for a sufficiently general type of control meshes. Anyone who tries to implement a subdivision scheme can observe that more often than not it is unclear how rules should be specified in certain cases (most commonly on boundaries and creases). Ad hoc solutions have to be used, which often have unexpected undesirable behavior. The lack of precise and complete definition makes it more difficult to exchange data between applications, reuse control meshes, and design new algorithms based on subdivision.

The difficulty in defining a reasonably complete set of subdivision rules is related to the fact that subdivision algorithms allow a large variety of data as input: an arbitrary polygonal or triangular mesh, possibly with boundary, marked edges, and vertices. Subdivision rules for the interior of a control mesh are well understood, while the boundary rules have received less 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. Often, only an approximate definition is required for the interior of the surface, whereas the boundary conditions may be significantly more restrictive. For example, it is often necessary to join several surfaces along their boundaries. Boundary subdivision rules lead to rules for sharp creases [8] and soft creases [3]. In addition to specifying the boundary or crease curves, it is often desirable to be able to specify tangent planes on the boundary; existing subdivision schemes do not allow to control 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 schemes of Catmull-Clark [2] and Loop [10]. The properties of our schemes were rigorously verified. We use a uniform approach to derive a set of rules, including new rules for concave corners, improved smooth boundary rules, new rules for tangent plane modification, and C^2 rules. While our approach is based on a number of known ideas, its advantage is that all desired features are handled in a unified framework.

Our approach to building a complete set of rules 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. The code implementing our algorithms is available on the Web[1].

2 Previous Work

A number of subdivision schemes have been proposed since Catmull and Clark introduced subdivision surfaces in 1978 [2]. A detailed survey of subdivision can be found in [1].

Theoretical analysis of subdivision rules was performed in [18, 15, 6, 19, 24, 23]. Most of this work has focused on closed surfaces; while the general theory does not impose symmetry restrictions on the subdivision rules, almost all theoretical analysis of specific schemes relies on the rotational symmetry of the subdivision rules and applies only to the interior rules.

Subdivision rules for Doo-Sabin dual surfaces for the boundary were discussed by Doo [4] and Nasri [12, 13, 11], but only partial theoretical analysis was performed. Our work builds on the work of Hoppe et al. [8] and partially on the ideas of Nasri [14].

To the best of our knowledge, the boundary subdivision rules proposed in work [8] are the only ones that result in provably C^1-continuous surfaces (the analysis can be found in Schweitzer [19]). 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.

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

Sederberg et al. [20] proposed a generalization of Catmull-Clark and Doo-Sabin subdivision rules that contains NURBS as a subset. For some applications it is important to include NURBS patches, however, the complexity of the algorithms is increased and the behavior of the surface near the extraordinary points becomes difficult

*biermann@mrl.nyu.edu
†adilev@math.tau.ac.il
‡dzorin@mrl.nyu.edu

[1] http://www.mrl.nyu.edu/biermann/sub

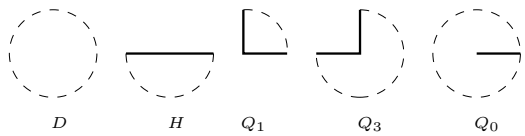


Figure 1: The charts for a surface with piecewise smooth boundary.

to analyze and predict. The smooth crease effects that are obtained by manipulating NURBS weights for subdivision surfaces can be achieved using an elegant technique proposed by DeRose et al. [3].

Our approach to C^2 subdivision is similar to the approach of [16].

Levin recently introduced a combined subdivision scheme which interpolates a network of curves [9]. There are two main distinctions between the present work and [9]. First, we are solving a different problem: rather than assuming that we are given a network of smooth curves that has to be interpolated, we assume only a discrete mesh with tags, which controls the behavior of our surface, but no interpolation is required. Second, Levin's combined subdivision schemes are an interesting new research direction; not much is known and understood about their behavior, especially on arbitrary meshes. In contrast, we focus on completing the subdivision toolbox with provably reliable tools.

Halstead et al. [7] describe a method of interpolating positions and normal direction on subdivision surfaces. However, this method involves the solution of a global system of equations, unlike our local subdivision rules.

3 Piecewise smooth surfaces

Piecewise smooth surfaces. Our goal is to design subdivision schemes for the class of *piecewise smooth surfaces*. This class includes common modeling primitives such as quadrilateral free-form patches with creases and corners. However, we exclude certain singularities (e.g., cone-like singularities and corners).

Here we give a somewhat informal description of piecewise-smooth surfaces, mathematical details will be presented elsewhere [25]. For simplicity, we consider only surfaces without self-intersection.

Recall that for a closed C^1-continuous surface in $\mathbf{R}^3$, each point has a neighborhood that can be smoothly deformed (that is, there is a C^1 map of maximal rank) into an open planar disk D. A surface with a *smooth boundary* can be described in a similar way, but neighborhoods of boundary points can be smoothly deformed into a half-disk H, with closed boundary (Figure 1). In order to allow *piecewise* smooth boundaries, we introduce two additional types of local charts: concave and convex corner charts, Q_3 and Q_1. We conclude that a C^1-continuous surface with piecewise smooth boundary looks locally like one of the domains D, H, Q_1, or Q_3. *Piecewise-smooth surfaces* are constructed out of surfaces with piecewise smooth boundaries joined together. If two surface patches have a common boundary, but different normal directions along the boundary, the resulting surface has a sharp crease.

We allow two adjacent smooth segments of a boundary to be joined, producing a crease ending in a *dart* (cf. [8]). 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.

It is important to observe that convex and concave corners, while being equivalent topologically, are not differentially equivalent. That is, there is no C^1 *nondegenerate* map from Q_1 to Q_3. Therefore, a single subdivision rule can not produce both types of corners [26]. 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 common rules

In this section, we demonstrate some problems of existing subdivision rules. We will see that not all piecewise-smooth surfaces can be adequately represented in these schemes.

Concave corners. Concave corners often arise in modeling tasks (e.g., surfaces with holes). In an attempt to model such a corner with subdivision surfaces, one might arrange the control mesh in a concave configuration and expect the surface to approximate the configuration. However, the corner rules of popular subdivision schemes (e.g., [8]) can only generate convex corners. If the control mesh is in a concave configuration, the rules force the surface to approach the corner from the outer, convex, side, causing the surface to develop a fold (Figure 2).

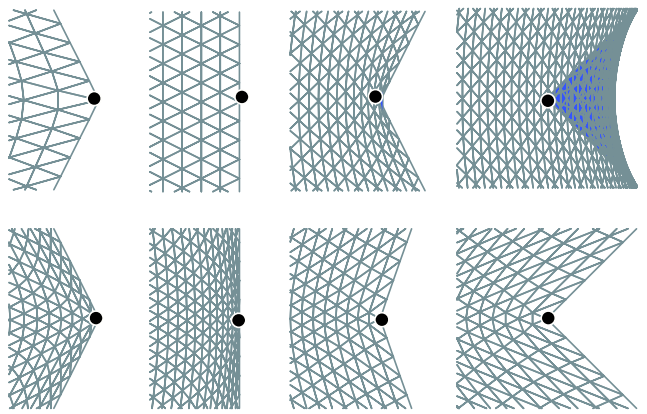


Figure 2: Upper row: behavior of a subdivision surface when rules of Hoppe et al. [8] are applied near a corner of the control mesh. As the corner of the control mesh is moved, the surface develops a fold. Lower row: our concave corner rules applied to the same mesh. The concave rules produce a small fold if applied to a convex control mesh configuration (not visible in the picture). For a concave configuration, our rule produces surfaces without folds.

Boundary rules. Hoppe et al. [8] observed that standard subdivision rules fail to produce smooth surfaces at extraordinary boundary vertices. They propose to change the subdivision scheme for the boundary curve in order to generate smooth surfaces. However, the boundary curve now depends on the interior of the control mesh. More specifically, the number of the interior vertices adjacent to each boundary vertex. This side effect is undesirable if one wants to join surfaces along their boundary curves: Two separate meshes might initially have the same boundary, but after subdivision a gap between the meshes can appear (Figure 6).

Moreover, even though the rules of [8] are formally smooth, they might produce undesirable sharp normal transitions if the control mesh is twisted (Figure 7).

5 Subdivision and eigenanalysis

In this section, we briefly state several facts of the theory of subdivision [1], which are helpful to understand the problems described above and our solutions.

Subdivision algorithms recursively refine a control mesh, recomputing vertex positions and inserting new vertices on edges (and possibly faces).

Our method of constructing subdivision rules is based on manipulating the eigenstructure of *subdivision matrices* associated with most common subdivision rules. This idea can be traced back to

[5]. Consider a vertex v, and let p be the vector of control points in a neighborhood of the vertex (Figure 3).

Let S be the matrix of subdivision coefficients relating the vector of control points p^m on subdivision level m to the vector of control points p^{m+1} on a similar neighborhood on the next subdivision level. Suppose the size of the matrix is N. Many properties of the subdivision scheme can be deduced from the eigenstructure of S. Let us decompose the vector of control points p with respect to the eigenbasis $\{x^i\}$, $i = 0..N-1$, of S, $p = \mathbf{a}_0 x^0 + \mathbf{a}_1 x^1 + \mathbf{a}_2 x^2 + \dots$ (it exists in the cases of importance to us).

Note that we decompose a vector of 3D points: the coefficients $\mathbf{a}_i$ are 3D vectors, which are componentwise multiplied with eigenvectors x^i.

We assume that the eigenvectors x^i are arranged in the order of non-increasing eigenvalues. For a convergent scheme, the first eigenvalue λ_0 is 1, and the eigenvector x_0 has all components equal to one; this is also required for invariance with respect to rigid and, more generally, arbitrary affine transformations.

Subdividing the surface m times means that the subdivision matrix is applied m times to the control point vector p.

$$S^m p = \lambda_0^m \mathbf{a}_0 x^0 + \lambda_1^m \mathbf{a}_1 x^1 + \lambda_2^m \mathbf{a}_2 x^2 + \cdots \quad \text{(Iterated Subdivision)}$$

If we further assume that λ_1 and λ_2 are real and equal, and $\lambda_1 = \lambda_2 = \lambda > |\lambda_3|$, we see from this formula that the vector of control points p^m can be approximated by $\mathbf{a}_0 x^0 + \lambda^m (\mathbf{a}_1 x^1 + \mathbf{a}_2 x^2)$; the rest of the terms decay to zero faster. If $\mathbf{a}_1 \times \mathbf{a}_2$ is not zero, then all of the control points p_i^m are close to the plane passing through $\mathbf{a}_0$ and spanned by vectors $\mathbf{a}_1$ and $\mathbf{a}_2$. As $m \to \infty$, the positions of all points converge to a_0.

This means that the limit position of the center vertex is $\mathbf{a}_0$; the tangent directions at this position are $\mathbf{a}_1$ and $\mathbf{a}_2$. We compute these values using the left eigenvectors of S (i.e., vectors l^i, satisfying $(l^i, x^i) = 1$ and $(l^i, x^j) = 0$ if $i \neq j$): $\mathbf{a}_i = (l^i, p)$.

These observations form the basis of our method: to ensure convergence to the tangent plane, we decrease the magnitudes of all eigenvalues except for those that correspond to the vectors $\mathbf{a}_1$, $\mathbf{a}_2$ spanning the desired tangent plane. We also modify the vectors $\mathbf{a}_1$ and $\mathbf{a}_2$ to change the direction of the normal. It should be noted that obtaining the correct spectrum of the subdivision matrix is not sufficient for smoothness analysis of subdivision; once our rules are formulated, we still have to prove that the resulting surfaces are C^1, using the characteristic map analysis [25].

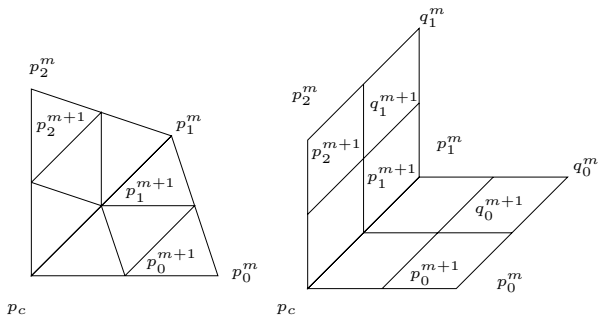


Figure 3: Neighborhoods of a vertex on different subdivision levels. The subdivision matrix relates the vector of control points p^m to the control points on the next level p^{m+1}. For a neighborhood of k triangles $p^m = \{p_c^m, p_0^m \dots p_{k-1}^m\}$, for k quadrilaterals $p^m = \{p_c^m, p_0^m \dots p_{k-1}^m, q_0^m \dots q_{k-1}^m\}$

6 Algorithm

6.1 Tagged meshes

Before describing our set of subdivision rules, we start with the description of the tagged meshes which our algorithms accept as input. We use these meshes to represent piecewise-smooth surfaces: edges and vertices of the mesh can be tagged to generate the singularities described in Section 3.

The complete list of tags is as follows. Edges can be tagged as *crease edges*. A vertex with incident crease edges receives one of the following tags:

- *crease vertex:* joins exactly two incident crease edges smoothly.
- *corner vertex:* connects two or more creases in a corner (convex or concave).
- *dart vertex:* causes the crease to blend smoothly into the surface.

We require that all edges on the boundary of the mesh are tagged as crease edges. Boundary vertices are tagged as corner or crease vertices.

Crease edges divide the mesh into separate patches, several of which can meet in a corner vertex. At a corner vertex, the creases meeting at that vertex separate the ring of triangles around the vertex into sectors. We label each sector of the mesh as *convex sector* or *concave sector* indicating how the surface should approach the corner.

The only restriction that we place on sector tags is that we require concave sectors to consist of at least two faces. An example of a tagged mesh is given in Figure 4.

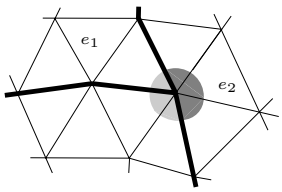


Figure 4: Crease edges meeting in a corner with two convex (light grey) and one concave (dark grey) sectors. Our subdivision scheme modifies the rules for edges incident to crease vertices (e.g., e_1) and corners (e.g. e_2).

In our implementation, the user applies the tags interactively, and the user interface prohibits an inconsistently tagged mesh (for example, there cannot be a corner vertex with some sector untagged). Also, the user can specify normal directions and *flatness parameters* for untagged vertices, crease vertices, and for each sector at a corner vertex. The flatness parameter determines how quickly the surface approaches the tangent plane in the neighborhood of a control point. This parameter is essential to our concave corner rules. Additionally, it improves the user control over the surface, for example, one can flatten a twist in the mesh (as shown in Figure 7). It is important to note however, that while manipulating these parameters is possible, it is not necessary: we provide default values reasonable for most situations (Section 6.2).

6.2 Subdivision rules

We describe our sets of rules for the triangular and quadrilateral schemes in parallel, as they are structurally very similar.

Our algorithm consists out of two stages, which, if desired, can be merged, but are conceptually easier to understand separately.

The first stage is a single iteration over the mesh during which we refine the position of existing vertices (vertex points) and insert new

vertices on edges (edge points). For the quadrilateral scheme, we also need to insert vertices in the centers of faces (face points). The first stage is similar to one subdivision step of standard algorithms, but the weights that we use are somewhat different. In the following we refer to the rules of Loop and Catmull-Clark as standard rules.

Vertex points. We apply the standard vertex rules to reposition untagged vertices and dart vertices. The new control point at a vertex is the weighted average of the control points in its neighborhood.

If a vertex has k adjacent polygons, then its new position is a combination of the old position with weight $5/8$ and of the sum all surrounding control points with weight $3/8k$, for $k \neq 3$. In case $k = 3$ we use a special set of coefficients with the weight of the central vertex equal to 7/16 [22]. For the quadrilateral scheme, the center vertex has weight $1 - \beta_1 - \beta_2$, while all adjacent vertices have weight β_1/k; the remaining vertices in the ring receive weight β_2/k with $\beta_1 = 3/(2k)$ and $\beta_2 = 1/(4k)$.

A crease vertex is refined as the average of its old position with weight $3/4$ and the two adjacent crease vertices with weight $1/8$ each. Corner vertices are interpolated.

Face points. For the quadrilateral scheme we insert a vertex at the centroid of each face; only one rule is necessary.

Edge points. This is the most complicated case. We choose the rule for an edge point depending on the tag of the edge and the tags of adjacent vertices and sectors. In the absence of tags, we apply the standard edge rules. The averaging masks are given in Figure 5.

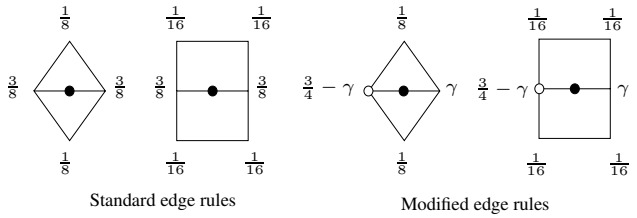


Figure 5: Edge rules for triangular and quadrilateral schemes. These rules apply to untagged edges. When both endpoints are untagged, we use standard rules. In case of a tagged endpoint we modify the rule such that the tagged endpoint (marked with a circle) receives coefficient $3/4 - \gamma$.

We insert a new vertex on a crease edge as the average of the two adjacent vertices.

The remaining case of an untagged edge e adjacent to a tagged vertex v is illustrated in Figure 4. We modify the standard edge rule in the following way: we parameterize the rule by θ_k, which depends on the adjacent vertex tag and sector tag. Let the vertices be labeled as in Figure 3, and let the position of the tagged endpoint be p_c^m, the other endpoint is p_i^m. We insert a vertex on the edge at position p_i^{m+1}. The edge rule for the triangular scheme is

$$p_i^{m+1} = (3/4 - \gamma)\, p_c^m + \gamma p_i^m + 1/8\, (p_{i-1}^m + p_{i+1}^m)\,.$$

We use a similar rule for the quadrilateral case:

$$p_i^{m+1} = (3/4 - \gamma)\, p_c^m + \gamma p_i^m + 1/16\, (p_{i-1}^m + p_{i+1}^m + q_{i-1}^m + q_i^m)\,.$$

The subdivision masks are illustrated in Figure 5. In each case γ is given in terms of parameter θ_k:

$$\begin{aligned} \gamma(\theta_k) &= 1/2 - 1/4 \cos\theta_k \quad \text{(triangular scheme)} \\ \gamma(\theta_k) &= 3/8 - 1/4 \cos\theta_k \quad \text{(quadrilateral scheme).} \end{aligned}$$

For a dart vertex v, we use $\theta_k = 2\pi/k$, where k is the total number of polygons adjacent to v. If v is a crease vertex, we use $\theta_k = \pi/k$, where k is the number of polygons adjacent to v in the sector of e.

At a corner vertex v we differentiate whether e is in a convex or concave sector. For a convex corner we use $\theta_k = \alpha/k$, where α is the angle between the two crease edges spanning the sector (k as above), for concave corners $\theta_k = (2\pi - \alpha)/k$.

6.3 Flatness and normal modification

The second stage of the algorithm is always applied at concave corner vertices and vertices with prescribed normals. It can be also applied at other boundary and interior vertices when it is desirable to increase flatness near a vertex or achieve C^2-continuity.

There are two slightly different types of position modifications performed at this stage: normal and flatness modification. Whenever we compute a vertex position in the neighborhood of a vertex subject to normal or flatness modification we compute the position using the rules above and modify it in a second step. The required eigenvectors for these modifications are listed in the appendix A.

Flatness modification. We observe that we can control how quickly the control points in a neighborhood converge towards the tangent plane. The equation for iterated subdivision suggests to accelerate the convergence by reducing eigenvalues $\lambda_i, i = 3 \ldots N-1$. We introduce a *flatness parameter* s and modify the subdivision rule to scale all eigenvalues except λ_0 and $\lambda = \lambda_1 = \lambda_2$ by factor $1 - s$. The vector of control points p after subdivision in a neighborhood of a point is modified as follows:

$$p^{\text{new}} = (1 - s)\, p + s \left(\mathbf{a}_0 x^0 + \mathbf{a}_1 x^1 + \mathbf{a}_2 x^2\right),$$

where $\mathbf{a}_i = (l^i, p)$, and $0 \leq s \leq 1$. Geometrically, the modified rule blends between control point positions before flatness modification and certain points in the tangent plane, which are typically close to the projection of the original control point. The limit position $\mathbf{a}_0$ of the center vertex remains unchanged.

The flatness modification is always applied at concave corner vertices; the default values for the parameter s is $s = 1 - 1/(2 + \cos\theta_k - \cos k\theta_k)$, which ensures that the surface is C^1 in this case. In other cases, s can be taken to be 0 by default.

C^2-modification. The flatness modification can be also used to make the subdivision scheme C^2, similar to the flat spot modifications [16]. It is known from the theory of subdivision that under certain conditions a scheme which is C^2 away from extraordinary vertices, generates surfaces which are C^2 at extraordinary vertices if all eigenvalues excluding 1 and λ are less than the squared subdominant eigenvalue. This can be easily achieved using flatness modification: s is taken to be less than $|\lambda|^2 / \max_{i>3} |\lambda_i|$. In general, values of s close to this quantity produce surfaces of better shape, but with greater curvature oscillations. It is worth noting that this approach has a fundamental problem: the resulting surface has zero curvature at the extraordinary vertex; the results of [17] indicate that for schemes with small support this is inevitable.

Normal modification. We introduce a similar modification, which allows one to interpolate given tangent and normal position at a vertex v. As above, we modify the control point positions in v's neighborhood after each subdivision step. In this case, the parameter t blends between the unmodified positions and positions in the prescribed tangent plane, while the limit position $\mathbf{a}_0$ of v remains unchanged.

For a prescribed tangent vector pair $\mathbf{a}_1'$ and $\mathbf{a}_2'$, we modify

$$p^{\text{new}} = p + t \left(\left(\mathbf{a}_1' - \mathbf{a}_1\right) x^1 + \left(\mathbf{a}_2' - \mathbf{a}_2\right) x^2\right);$$

where $\mathbf{a}_i = (l^i, p)$ and $0 \leq t \leq 1$. In case of a prescribed normal direction n we compute the tangent vectors as $\mathbf{a}_i' = \mathbf{a}_i - (\mathbf{a}_i, n)n$.

We observe that the subdivision rules are no longer applied to each coordinate of the control points separately; rather, the whole 3D vector is required. We can think of this as a generalized form of subdivision, where the coefficients are matrices rather than scalars. Thus, a control point position p_i^{m+1} in a neighborhood with prescribed normal n on level $m+1$ can be explicitly expressed as

$$p_i^{m+1} = \sum_j p_j^m \left(s_{ij}\mathbf{Id} - t\left(\sum_k x_i^1 l_k^1 s_{kj} + x_i^2 l_k^2 s_{kj} \right) n^T n \right)$$

where s_{ij} are entries of the original subdivision matrix S and **Id** the 3×3 identity matrix. It should be noted that our analysis in [25] applies only to the case $t = 1$, which we use as a default value; the analysis of the general case is still an open question.

7 Discussion

We have presented a number of simple extensions to the standard Catmull-Clark and Loop subdivision schemes that resolve some problems with existing rules.

Our rules are designed to coincide with cubic endpoint interpolating B-splines rules along a crease. As a consequence, the generated crease curves depend only on the crease control points. Therefore, it is possible to modify the interior of a surface patch without any effect on the bounding crease curves; moreover, one can join piecewise-smooth surfaces without gaps and combine them with other surface representations supporting B-spline boundaries.

A complete C^1-continuity analysis of our subdivision rules is outside of the scope of this paper, and will be given elsewhere [25]. Here we describe only the intuition behind our construction.

We can understand the behavior of the surface in a neighborhood of a corner or crease vertex from the eigenstructure of the corresponding subdivision matrix.

If we apply the standard rules in the neighborhood of a crease vertex, the eigenvalue $1/2$ corresponding to the tangent vector of the crease is not subdominant. As a result, the surface contracts at a different rate from the crease, leading to a degenerate configuration without tangent plane (Figure 2). The situation for corner vertices is similar as both tangent vectors are determined from crease curve segments with eigenvalue $1/2$.

Our subdivision rules ensure that $1/2$ is the subdominant eigenvalue in both cases. It it not difficult to see that $1/2$ is an eigenvalue: Consider a planar fan of k congruent polygons, where each polygon contributes an angle θ_k to the total angle $\theta = k\theta_k$. If we treat this configuration as a crease or corner neighborhood and apply our modified subdivision rules, then the center vertex does not change its position, and for each adjacent edge we insert a vertex at exactly the midpoint. Thus, the configuration is scaled down by a factor of $1/2$, i.e., $1/2$ is an eigenvalue.

It turns out that $\lambda = 1/2$ is indeed subdominant for crease vertices and convex corners. For concave corners we ensure subdominance by reducing all other eigenvalues (except $\lambda_0 = 1$) using the flatness modification with parameter s satisfying $(1-s)(2 + \cos\theta_k - \cos k\theta_k) < 2$. Figure 11 demonstrates how the flatness modification pulls the neighborhood of a convex corner into its tangent plane.

Our implementation of the rules is available on the Web. We have also developed explicit evaluation rules for our schemes, extending [21].

8 Results and Conclusions

Surfaces with creases and corners of various types are illustrated in Figures 12 and 13(b). All the surfaces in Figure 12 are generated from the same control mesh by applying different tags. Note how convex and concave sectors meet along the crease of the torus.

Figures 8 and 10 demonstrate normal interpolation for boundary, corner and interior vertices; directions of normals are adjusted to obtain desired shapes without modifying the control mesh. Other applications are possible: we have applied normal modification to create certain surface characteristics: randomly perturbing the top-level normals produces a wavy doughnut from a torus-like control mesh; perturbing normals on the first subdivision levels creates a noisy doughnut (Figure 13(c) and (d)).

Conclusions 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 boundary and creases and provides additional controls for surface modeling.

Even though the class of surfaces considered in this paper is quite general, we have excluded many types of surface singularities. Future work might explore which other singularities are useful for modeling purpose and how to construct subdivision rules to create such features.

9 Acknowledgments

We are greatly indebted to Peter Schröder for his support and suggestions, and for trying out the schemes and discovering numerous hard-to find typos in the formulas. This work has its origin in discussions with Tom Duchamp. We would like to thank the anonymous reviewers for their comments.

A portion of this work was supported by NSF award ACI-9978147.

References

[1] Subdivision for modeling and animation. SIGGRAPH 2000 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 C^1 subdivision surfaces. *Computer Aided Geometric Design*, 16(4):223–247, 1999.

[7] Mark Halstead, Michael Kass, and Tony DeRose. Efficient, fair interpolation using Catmull-Clark surfaces. In *Computer Graphics Proceedings*, Annual Conference Series, pages 35–44. ACM Siggraph, 1993.

[8] 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.

[9] Adi Levin. Interpolating nets of curves by smooth subdivision surfaces. In Alyn Rockwood, editor, *SIGGRAPH 99 Conference Proceedings*, Annual Conference Series, pages 57–64. Addison Wesley, 1999.

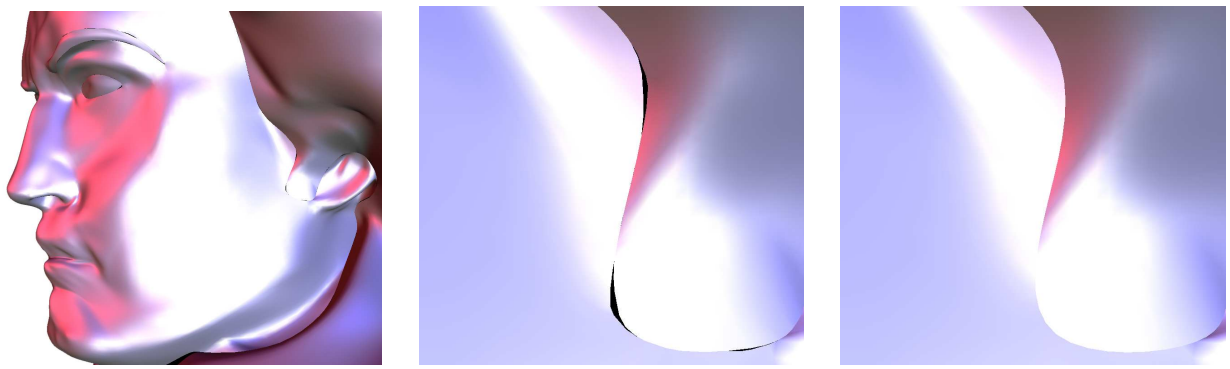
[10] Charles Loop. Smooth subdivision surfaces based on triangles. Master's thesis, University of Utah, Department of Mathematics, 1987.

[11] A. Nasri. Interpolation of open B-spline curves by recursive subdivision surfaces. In Tim Goodman and Ralph Martin, editors, *Mathematics of Surfaces VII*, pages 173 –188. Institute of mathematics and its applications, Information Geometers, 1997.

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

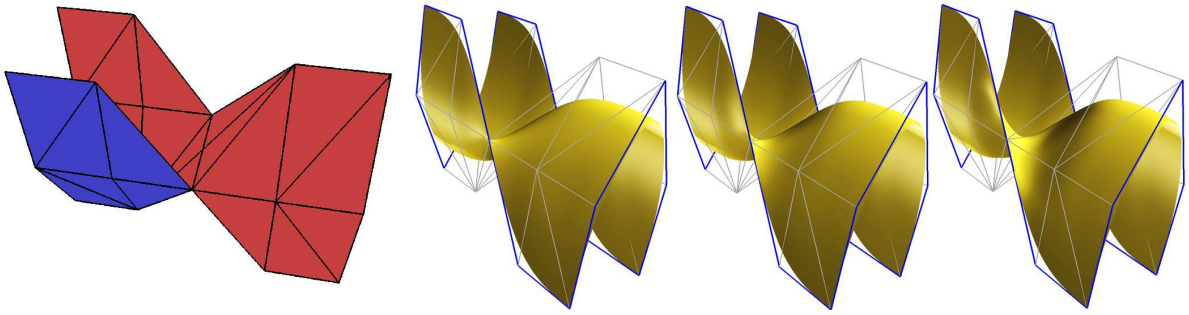
[13] Ahmad H. Nasri. Boundary corner control in recursive subdivision surfaces. *Computer Aided Design*, 23(6):405–410, 1991.

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



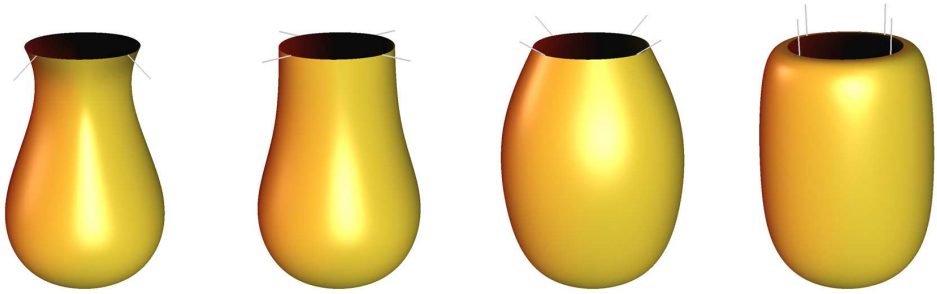
(a) (b) (c)

Figure 6: Subdivision on meshes with boundaries: Beethoven's face and hair are modeled as separate meshes with identical boundaries. (a) and (b): the rules of [8] result in a gap between the surfaces due to extraordinary vertices. (b) A close-up on the gaps at the ear. (c) With our rules no gap is created.



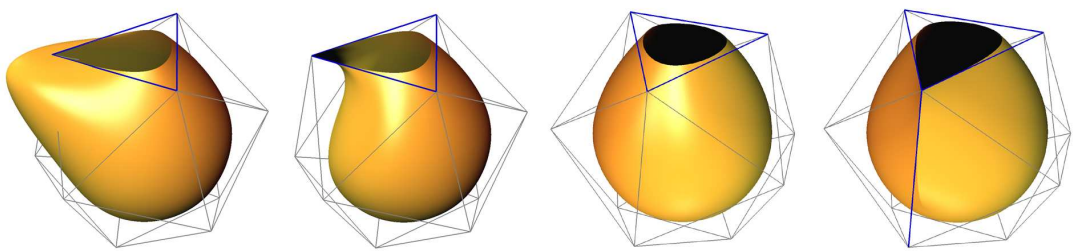
(a) (b) (c) (d)

Figure 7: (a) Control mesh with a twist on the boundary. (b) Normal varies rapidly near the point although the surface is formally smooth: there is a single bright spot on the front-facing boundary. (c), (d) Our algorithm reduces the variation: the highlights become larger.



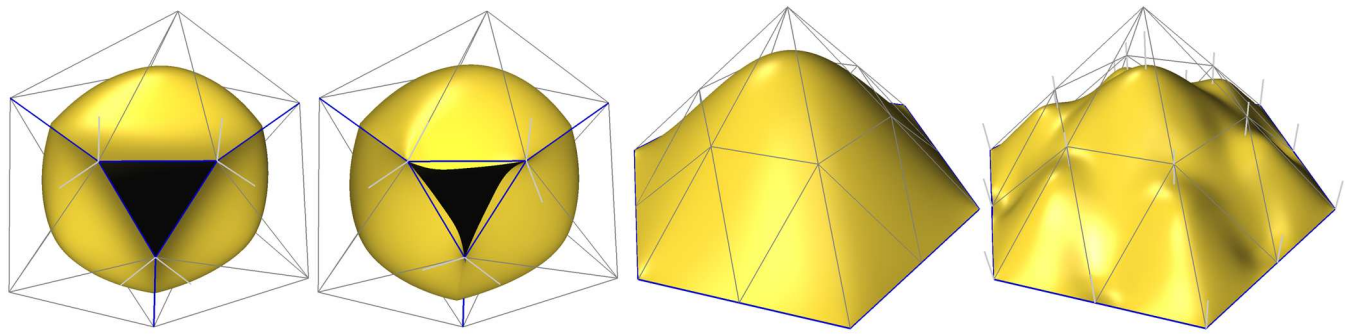
(a) (b) (c) (d)

Figure 8: Normal interpolation for quadrilateral subdivision. Prescribed directions: (a) tilted downwards, (b) horizontal, (c) no modification, (d) vertical.



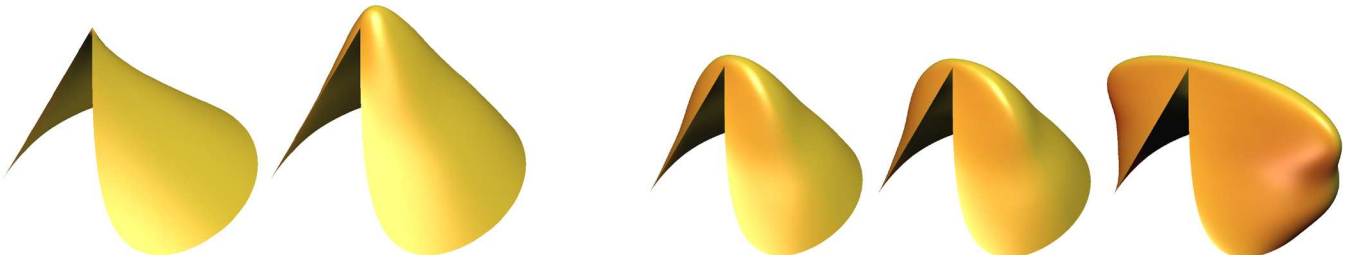
(a) (b) (c) (d)

Figure 9: Features: (a) concave corner, (b) convex corner, (c) smooth crease, (d) corner with two convex sectors.



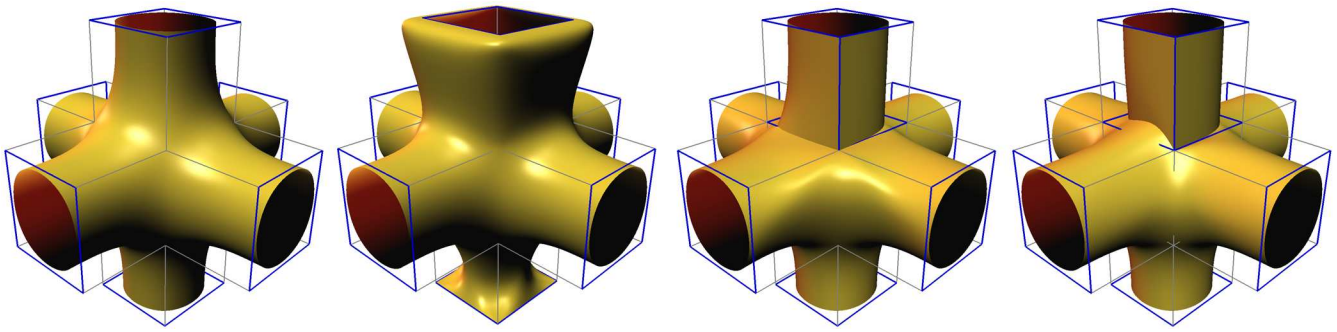
(a) (b) (c) (d)

Figure 10: Normal interpolation. (a) Surface with convex corners. (b) Prescribed directions: at each corner we tilt the normal for one surface sector slightly inwards. (c) Smooth surface. (d) Same control mesh but all normals vertical.



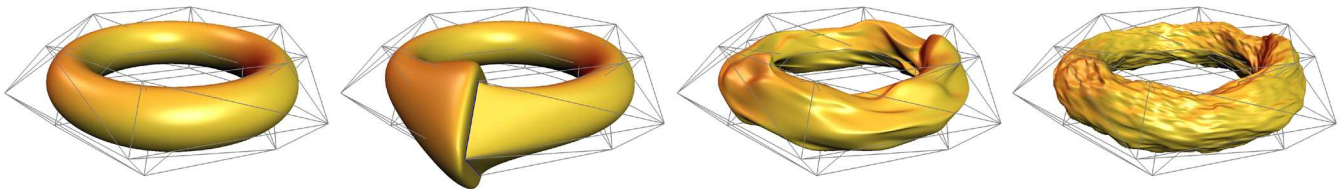
(a) (b) (c) (d) (e)

Figure 11: Concave corner rules. (a) A corner without flatness modification. (b) Flatness modification lifts the surface into its tangent plane. (c-e) The corner shape for different values of θ_k.



(a) (b) (c) (d)

Figure 12: Surface manipulation with corners. (a) Smooth boundary curves. (b) Concave corners on top, convex corners on bottom. (c) Corners with convex and concave sectors. (d) Creases and corners as for (c) but with prescribed normal direction on concave sectors.



(a) (b) (c) (d)

Figure 13: Manipulating a torus. (a) The original surface. (b) A surface with creases and convex/concave corners. (c) Wavy torus: we deform the torus by randomly perturbing normals of the control mesh. (d) Noisy torus: we perturb the normals on the first four subdivision levels.

[15] Jörg Peters and Ulrich Reif. Analysis of algorithms generalizing B-spline subdivision. *SIAM Journal on Numerical Analysis*, 35(2):728–748 (electronic), 1998.

[16] H. Prautzsch and G. Umlauf. A G^2-subdivision algorithm. In *Geometric modelling (Dagstuhl, 1996)*, pages 217–224. Springer, Vienna, 1998.

[17] Hartmut Prautzsch and Ulrich Reif. Degree estimates for C^k-piecewise polynomial subdivision surfaces. *Adv. Comput. Math.*, 10(2):209–217, 1999.

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

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

[20] 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.

[21] Jos Stam. Exact evaluation of Catmull-Clark subdivision surfaces at arbitrary parameter values. In *SIGGRAPH 98 Converence Proceedings*, Annual Conference Series, pages 395–404. Addison Wesley.

[22] Joe Warren. Subdivision methods for geometric design. Unpublished manuscript, November 1995.

[23] Denis Zorin. A method for analysis of C^1-continuity of subdivision surfaces. 1998. submitted to SIAM Jornal of Numerical Analysis.

[24] Denis Zorin. Smoothness of subdivision on irregular meshes. *Constructive Approximation*, 16(3), 2000.

[25] Denis Zorin, Tom Duchamp, and H. Biermann. Smoothness of subdivision surfaces on the boundary. Technical report, New York University, Dept. of Computer Scinece, 2000.

[26] 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 left and right eigenvectors necessary for the subdivision rules described above. Recall that the eigenvector coefficients are applied to a control points of a polygon ring/fan. A subscript c denotes the coefficient corresponding to the center vertex. For the quadrilateral scheme, we mark edgepoint coefficients with subscript p and facepoint coefficients with q.

We define the *degree* of a vertex as the number of polygons asjacent to this vertex; note that this definition is different from the standard one (the number of incident edges) for boundary vertices. The *crease degree* is the number of polygons adjacent to a crease or corner vertex with respect to a specific sector.

Also, recall that dominant right eigenvector x^0 is the vector consisting of ones.

Loop scheme.

- Interior vertex of degree k. In all cases i is in the range $0 \dots k-1$, and $\theta_k = 2\pi/k$.

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

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

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

- Smooth crease vertex of crease degree k. Let $\theta_k = \pi/k$; then

$$l_c^0 = 2/3, \quad l_1^0 = l_k^0 = 1/6, \quad l_i^0 = 0, \quad i = 1 \dots k-1$$

For $k = 1$,

$$x_c^1 = -1/3,\ x_1^1 = 2/3,\ x_2^1 = 2/3, \quad x_c^2 = 0,\ x_1^2 = 1,\ x_2^2 = -1$$

$$l_c^1 = -1,\ l_1^1 = 1/2,\ l_2^1 = 1/2, \quad l_c^2 = 0,\ l_1^2 = 1/2,\ l_2^2 = -1/2;$$

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

$$x_i^1 = \sin i\theta_k, \quad x_i^2 = \cos i\theta_k, \quad i = 0 \dots k$$

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

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

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

$$l_i^2 = \frac{2}{k}\sin i\theta_k, \quad i = 1 \dots k-1$$

where

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

$$\sigma_1 = \frac{\sin\theta_k}{1-\cos\theta_k} \qquad \sigma_3 = \frac{\cos\frac{k\zeta}{2}\sin\theta_k}{\cos\zeta - \cos\theta_k}$$

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

- Convex/concave corner vertex of crease degree k with parameter θ_k. Let $\theta = k\theta_k$.

$$l_c^0 = 1, \quad l_i^0 = 0, \quad i = 0 \dots k$$

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

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

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

Catmull-Clark scheme.

- Interior vertex of degree k. Let $\theta_k = 2\pi/k$ and i from 0 to $k-1$.

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

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

$$x_{pi}^1 = \frac{1}{\sigma}\sin i\theta_k \quad x_{pi}^2 = \frac{1}{\sigma}\cos i\theta_k$$

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

$$l_{pi}^1 = 4\sin i\theta_k \quad l_{pi}^2 = 4\cos i\theta_k$$

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

where $\lambda = 5/16 + 1/16\left(\cos\theta_k + \cos\theta_k/2\sqrt{9 + \cos 2\theta_k}\right)$, and $\sigma = k(2 + \frac{1+\cos\theta_k}{(4\lambda-1)^2})$.

- Smooth crease vertex of crease degree k. Let $\theta_k = \pi/k$.

$$l_c^0 = 2/3,\ l_{p1}^0 = l_{pk}^0 = 1/6,\ l_{q0}^0 = l_{qi}^0 = l_{pi}^0 = 0,\ i = 1 \dots k-1$$

For $k = 1$,

$$\begin{array}{llll} x_c^1 = 1/18, & x_{p0}^1 = -2/18, & x_{p1}^1 = -2/18, & x_{q0}^1 = -5/18 \\ x_c^2 = 0, & x_{p0}^2 = -1/2, & x_{p1}^2 = 1/2, & x_{q0}^2 = 0 \\ l_c^1 = 6, & l_{p0}^1 = -3, & l_{p1}^1 = -3, & l_{q0}^1 = 0 \\ l_c^2 = 0, & l_{p0}^2 = -1, & l_{p1}^2 = 1, & l_{q0}^2 = 0; \end{array}$$

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

$$x_{pi}^1 = \sin i\theta_k, \quad x_{pi}^2 = \cos i\theta_k, \quad i = 0 \dots k$$

$$x_{qi}^1 = \sin i\theta_k + \sin(i+1)\theta_k, \quad i = 0 \dots k-1$$

$$x_{qi}^2 = \cos i\theta_k + \cos(i+1)\theta_k, \quad i = 0 \dots k-1$$

$$l_{p0}^2 = 1/2, \quad l_{pk}^2 = -1/2, \quad l_{q0}^2 = l_{qi}^2 = l_{pi}^2 = 0, \quad i = 1 \dots k-1$$

$$l_c^1 = 4R(\cos\theta_k - 1), \quad l_{p0}^1 = l_{pk}^1 = -R(1 + 2\cos\theta_k)$$

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

$$l_{qi}^1 = \frac{4(\sin i\theta_k + \sin(i+1)\theta_k)}{(3+\cos\theta_k)k}, \quad i = 0 \dots k-1$$

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

- Convex/concave corner vertex of crease degree k with parameter θ_k. Let $\theta = k\theta_k$. Left eigenvectors are the same as for Loop with zeroes everywhere except l_c, l_{p0} and l_{pk}.

$$x_{pi}^1 = \frac{\sin i\theta_k}{\sin\theta}, \quad x_{pi}^2 = \frac{\sin(k-i)\theta_k}{\sin\theta}, \quad i = 0 \dots k$$

$$x_{qi}^1 = \frac{\sin i\theta_k + \sin(i+1)\theta_k}{\sin\theta}, \quad i = 0 \dots k-1$$

$$x_{qi}^2 = \frac{\sin(k-i)\theta_k + \sin(k-i-1)\theta_k}{\sin\theta}, \quad i = 0 \dots k-1$$