Nonmanifold Subdivision

Lexing Ying Denis Zorin
New York University*

Abstract

Commonly-used subdivision schemes require manifold control meshes and produce manifold surfaces. However, it is often necessary to model nonmanifold surfaces, such as several surface patches meeting at a common boundary.

In this paper, we describe a subdivision algorithm that makes it possible to model nonmanifold surfaces. Any triangle mesh, subject only to the restriction that no two vertices of any triangle coincide, can serve as an input to the algorithm. Resulting surfaces consist of collections of manifold patches joined along nonmanifold curves and vertices. If desired, constraints may be imposed on the tangent planes of manifold patches sharing a curve or a vertex.

The algorithm is an extension of a well-known Loop subdivision scheme, and uses techniques developed for piecewise smooth surfaces.

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

Additional Keywords: Subdivision surfaces, Nonmanifold surfaces, Geometric modeling.

1 Introduction

A crucial advantage of subdivision algorithms is their ability to generate a smooth surface from a broad class of control meshes. However, all existing subdivision schemes are designed for generating manifold surfaces. At the same time, many objects we may want to model are best represented by surfaces with nonmanifold topology. A surface has nonmanifold topology if at some points of the surface no neighborhood, no matter how small, can be continuously deformed into a disk. Typically, this occurs if the surface consists of several patches sharing a boundary. Examples of such surfaces include models of biological structures (heart with ventricles separated by a wall, fish with fins) as well as artificial objects (plane with wings, bag with several compartments). One of the motivations of this work was the need to produce models for heart simulations using immersed boundary method, as described in Section 4.

General polygonal or triangle meshes can be used to model such objects, if no requirements are imposed on the ways the faces of the mesh can share vertices or edges. However, the problem becomes more difficult if we wish to model the surfaces using higher order primitives, subdivision surfaces in particular: existing algorithms require the control mesh to be a topological manifold. It is possible to split such meshes into manifold pieces and treat each as a separate mesh. However, with this approach it is difficult to ensure that the pieces do not separate. Furthermore, it might be necessary to impose tangent plane continuity constraints on the joints between manifold pieces; such constraints are also difficult to maintain if the surface is defined by a collection of separate meshes.

Our approach is to extend existing subdivision rules to handle a more general class of control meshes. In fact, we allow any triangle mesh as input, as long as for any polygon no two vertices of this polygon coincide. Our scheme extends the well-known Loop subdivision scheme [8], but the same approach can be applied with few changes to extend any subdivision scheme to nonmanifold control meshes. Additional data assigned to vertices and edges of the mesh allows one to specify how patches are joined together.

Our main contributions include extending piecewise smooth subdivision rules to nonmanifold surfaces and new rules for tangent plane continuous interpolation of nonmanifold curves, an essential tool for modeling piecewise smooth nonmanifold surfaces.

We should note that nonmanifold surfaces form a very broad class; we restrict our attention to surfaces which have well-defined tangent planes everywhere, possibly several tangent planes at singular features such as nonmanifold vertices and curves. For such surfaces, at any point behavior can be described up to the first order by a collection of tangent planes of manifold surface patches meeting at the point. Thus, if we can ensure that a certain part of the surface has prescribed tangent plane at a point, we should be able to model any desired first order surface behavior. A limitation of our approach is that we do not include surfaces with no finite set of tangent planes defined at some points, e.g. a cone or several cones attached at their apexes. While including certain categories of cones would be easy, it is unclear at this point which types of cones are useful in practice; we leave this as a subject for future research.

Related Work. A recent paper [1] discussed methods to generate surfaces with piecewise smooth boundaries. Our work is based on the results of that paper. The crucial difference is the way the rules are applied and addition of special rules for tangent plane interpolation along a curve. Rules for tangent plane interpolation along a curve introduced in this paper are close in spirit to the work of Levin [6, 7]. The paper of Hubeli and Gross [4] considers the problem of fairing arbitrary nonmanifold meshes. Their approach to defining nonmanifolds is slightly different: they consider nonmanifold surfaces formed as unions of intersecting manifold surfaces. Topologically these classes are the same. Our work is complementary to [4]: rather

*{lexing,dzorin}@mrl.nyu.edu

than fairing existing fine meshes, we construct smooth surfaces from an arbitrary coarse control mesh.

The paper by Nasri [10] studies a related problem of interpolating networks of curves by dual subdivision surfaces; as we explain below, curve interpolation plays an important role in our construction. We consider primal rather than dual subdivision and our approach is more compatible with Loop and Catmull-Clark schemes.

Nonmanifold objects are widely studied in solid modeling literature [9, 12, 13, 2]. Other related work includes nonmanifold mesh compression [3] and nonmanifold simplification e.g. [11].

2 Nonmanifold Surfaces

We start with introducing notation for meshes with nonmanifold topology. For any vertex v of the mesh let $N_1(v)$ (1-neighborhood of v) be the set of all triangles sharing v. We call v a *manifold vertex*, if the triangles of $N_1(v)$ can be ordered so that two sequential triangles share one edge ending at v. The first and the last triangles may also share one edge, in which case v is an *interior* vertex; otherwise v is a *boundary* vertex. An edge of the mesh is called *manifold* if it is shared by two triangles of the mesh, or belongs to a single triangle; in the latter case it is called *boundary.*

We call all boundary and nonmanifold edges *singular*, and similarly *singular vertices* are the nonmanifold and boundary vertices. A *flower* F at a vertex v is a maximal

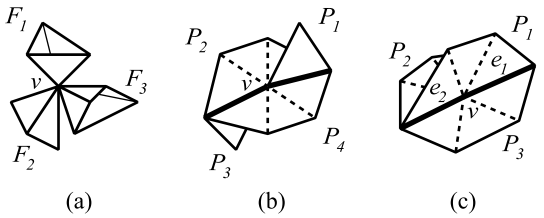


Figure 1: (a) Three flowers F_1, F_2 and F_3 at vertex v. (b) Four petals P_1, P_2, P_3 and P_4 at vertex v. (c) v is a simple nonmanifold vertex with 3 petals P_1, P_2 and P_3 which share two nonmanifold edges e_1 and e_2.

subset of $N_1(v)$ such that for any two triangles in F there is a chain of triangles in $N_1(v)$ connecting them such that any two sequential triangles in the chain share an edge. A *petal* P of a flower F is a maximal subset of F, such that any two triangles in P are connected by a chain of triangles in F such that any two consecutive triangles in the chain share a manifold edge (Figure 1(b)).

Thus, for a triangle mesh the 1-neighborhood of a vertex v consists of a collection of flowers; each flower, in turn, consists of a number of petals, bounded by singular edges incident on v.

Among nonmanifold vertices it is convenient to distinguish a special type of vertices which we call *simple.* A simple nonmanifold vertex v has a single flower in $N_1(v)$, and there are exactly two singular edges incident on v. It follows that all petals share these two edges (Figure 1(c)). All other nonmanifold vertices are called *complex.*

It follows from the definition of petals that at any vertex v, $N_1(v)$ can be decomposed into manifold petals at any vertex. Each petal can belong to one of the three types shown in Figure 2, depending on the number of singular edges it contains; we will call these types closed, dart and open petals.

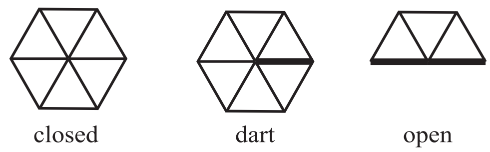


Figure 2: Three types of petals. The closed petal has no singular edges, the dart petal has one and the open petal has two. Singular edges are shown as dark edges.

Subdivision of nonmanifold meshes. To define a subdivision scheme we need to define two types of rules: connectivity refinement rule, which tells us how to change the connectivity of the mesh, and geometric rule, which tells us how to compute positions of control points. Standard quadrisection refinement rules work without any changes on nonmanifold meshes. To choose geometric rules, we need to define when nonmanifold surfaces are considered smooth. As there is no conventional definition, we choose one that we believe to be intuitively natural and sufficiently flexible for geometric modeling purposes.

The surface obtained by subdivision can be regarded as parameterized on the control mesh. Petals, singular curves and singular points on the surface are images of petals, singular edges and singular vertices respectively on the control mesh.

We observe that petals are not much more complex than local charts for a topological manifold with boundary; instead of two chart types we have three (the dart chart is added). It appears to be natural to treat petals separately in the same way we treat neighborhoods of points on manifold surfaces. Thus, a nonmanifold vertex has a collection of neighborhoods associated with it, and we say that a surface is smooth if each of these neighborhoods has a regular parameterization over one of the standard charts used for piecewise smooth surfaces (Figure 3).

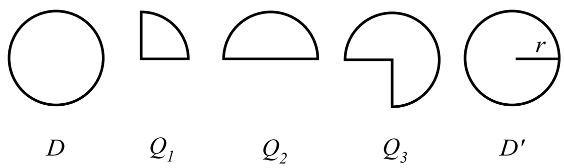


Figure 3: Domains for local parameterizations.

We assign the parameterizations as follows:
- closed petals are parameterized over the disk D;
- dart petals are parameterized over the disk D' with a singular radius r;
- open petals of simple singular vertices are parameterized over the half-disk Q_2;
- open petals of complex singular vertices can be parameterized over Q_1, Q_2 or Q_3.

These constraints immediately suggest how a simple subdivision scheme can be defined for nonmanifold meshes, using the rules for piecewise smooth subdivision surfaces described in [1]. We assume that the ambiguity for open petals of complex singular vertices is resolved by user choice of tags (convex corner, concave corner, smooth); if no tag is chosen, we assume that the petal is a convex corner. Note that two petals sharing two singular edges can be either both corner or both smooth. If one is concave, the other can be either convex or concave.

Nonmanifold subdivision scheme. We start with a summary of the rules; detailed formulas are provided below and in Figure 4. Vertex rules are as follows:

- for simple singular vertices, use cubic B-spline subdivision rules.
- for complex singular vertices, use corner rules, i.e. the vertex remains fixed, the user specified tag is used to decide the choice between convex and concave corner rule.
- for regular vertices, use standard Loop rules.

The edge rules follow the rules for piecewise smooth surfaces, with sectors replaced by petals, with one exception at the closed petal:

- for singular edges, use the standard cubic B-spline subdivision rule (midpoint subdivision).
- for nonsingular edges of a corner or dart petal, use corresponding piecewise surface rules.
- for nonsingular edges which are parts of a closed petal for a nonmanifold vertex, use a special *fixed disk rule* described below.
- in all other cases, use standard Loop rules.

It should be noted that on the first subdivision step a nonsingular edge can be contained in two different petals if both endpoints are singular vertices. In this case we simply use the average of the conflicting positions; after the first subdivision step this is no longer possible for both its subedges.

Edge rules in detail. As we have mentioned, edge subdivision rules coincide with piecewise smooth subdivision surface subdivision rules with the exception of an additional fixed disk rule. Here we briefly review these rules for completeness; for details, see [1].

The rules consist of two stages: subdivision and flatness modification. The stencils for the vertex rules are shown in Figure 4.

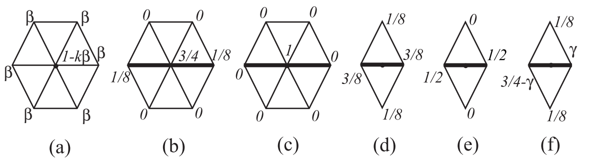


Figure 4: Vertex rules: (a) Standard Loop rule, k is the valence of center vertex. If $k \neq 3$, $\beta = 3/8k$, if $k = 3$, $\beta = 3/16$. (b) Uniform cubic spline rule. (c) Interpolation rule. Edge rules: (d) Standard Loop rule. (e) Uniform cubic spline rule. (f) modified edge rule, the center vertex of the petal has coefficient $3/4 - \gamma$. θ is the parameter for different cases, values are specified in the text. In the picture $\gamma = 1/2 - 1/4\cos\theta$.

Stencils used at the first stage of edge subdivision rules are shown in Figure 4 (d) to (f). Suppose k is the valence of the considered petal, the values of θ for dart, convex corner and concave corner rules are $2\pi/k$, α/k and α/k respectively, where in the last two cases, α is the angle between two singular curves of the petal. For smooth open petal rule (when two singular curves bounding the petal meet smoothly) we use $\theta = \pi/k$. For the fixed disk rule, the new rule introduced for closed petals at nonmanifold vertices, the standard Loop rule is used at the first stage ($\theta = \pi/3$).

Flatness modification. The second stage for edge rules is flatness modification, which is required for concave corner, dart and fixed disk rules to ensure C^1-continuity. Flatness, tangent and normal modification are controlled by a parameter s, which determines how fast the surface approaches the tangent plane.

To formulate the flatness modification rules we need additional notation. Consider a petal at a vertex v. Let p^m be the vector of control points on subdivision level m which are in $N_1(v)$ and also contained in the petal.

Subdivision matrix S is the matrix of subdivision coefficients which relates the vector of control points p^m at level m to the vector of control points p^{m+1} at level $m+1$: $p^{m+1} = Sp^m$. Let x^i be right eigenvectors of S, and λ_i corresponding eigenvalues. Then we can write $p = \sum_{i=0}^{N-1} a_i x^i$ Notice each a_i is a vector in R^3 computed as $a_i = (l^i, p)$, where l^i is the left eigenvector of S. The left eigenvectors for all cases excluding fixed disk rules are provided in [1]. The vectors for this rule are given in the appendix. In order for the subdivision scheme to be convergent, λ_0 is 1 and x_0 is a vector including all ones.

The flatness modification ensuring C^1 continuity is given by

$$p^{new} = (1-s)p + s(a_0 x^0 + a_1 x^1 + a_2 x^2).$$

In the case of concave corner rules, in order to ensure C^1-continuity, s has to satisfy $(1-s)(3/8 + (1/4)(\cos(\pi/k) - \cos(\theta))) < 1/2)$. Clearly, choosing $s = 1/2$ is always sufficient. We refer the reader to [1] for details.

However, the subdivision scheme that we have just described is not sufficiently flexible. While we ensure that each petal is a piecewise smooth surface, we have no means of making two petals share a tangent plane along a curve or at a point, which is a natural operation (Figure 5, Figure 12).

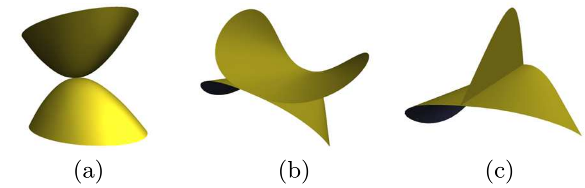


Figure 5: (a) Two tangential patches at a vertex. (b) Four patches share a tangent plane along a common edge. (c) A patch attached to a smooth surface.

3 Tangent Plane Constraints

Our initial scheme described above needs to be extended to handle constraints between patches.

Enforcing a common normal for two petals joined at a point is straightforward; normal modification of [1] can be used for this.

Ensuring two patches of the surface meeting at curve to have a common normal along the curve is more difficult.

Our approach is to use rules that allow one to prescribe the normal to the surface everywhere on the common curve explicitly; if such rules are used, one can guarantee that two patches share a singular curve and have common normal along that curve, and still evaluate subdivision on each patch separately. An additional advantage of this approach is that any number of patches can be made tangential at a nonmanifold curve. It turns out that *tangent modification*, similar to flatness modification described above, can be used to force prescribed normal along a singular edge, if applied at all simple singular vertices on that edge.

The rules that we use are similar to the approach of Levin [6, 7]; however, they do not require explicit use of the parameterization of the singular curve and do not place any restrictions on the valence of vertices on the curve.

Prescribed tangent and normal data. Before describing our rules that enforce prescribed normal along singular curves and at singular vertices, we describe the tangent data that can be associated with the mesh vertices to control this behavior. We take a somewhat more general approach than required to handle the cases listed above, and allow the user to prescribe a complete set of tangent planes for any singular point.

More specifically, the following data can be specified:

- tangents for any singular edges incident at complex singular vertices;
- additional "interior" tangents for any dart and smooth open petals;
- normals for any closed petals.
- Singular edges can be tagged; prescribed tangent planes are interpolated along these edges.

One can easily see that at a singular vertex if all the data is specified, the tangent planes are determined for each petal. Indeed, for a corner petal the normal is defined by the tangents to the singular curves bounding the petal. For a smooth open petal an extra interior tangent has to be specified because the singular curves have the same tangent at the vertex. Similarly, for a dart vertex an interior tangent has to be specified as there is only one singular curve. Finally, the normal can be specified independently for any closed petal.

In addition to vertex data, some singular edges are tagged. For these edges the tangent plane should be interpolated along a singular edge independently of the control point positions away from the edge. When we subdivide the edge, we also use midpoint subdivision to interpolate interior tangents (this is sufficient for C^1 but not C^2 continuity). Interpolated interior tangent may have a singularity, but this happens only if two prescribed tangents at adjacent vertices have strictly opposite directions.

This mechanism provides a general way of controlling the tangent plane along a singular curve; for example, by prescribing opposite interior tangents for two patches at two adjacent vertices, we ensure that near the edge the two patches will form a C^1-continuous surface with the singular curve embedded in the surface (Figure 6(a)).

To ensure that the prescribed and interpolated tangents are actually tangent to the surface, we use two rules taken from [1]: tangent and normal modification.

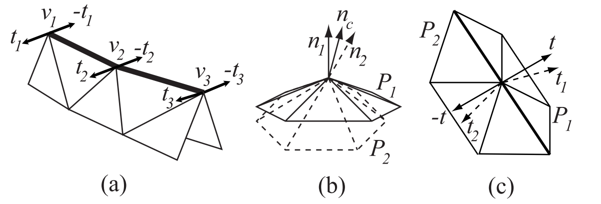


Figure 6: (a) To force two patches to share the same tangent plane along a singular curve, we specify the interior tangents at all vertices of the curve. Note that at any v_i on the curve, opposite interior tangents t_i and $-t_i$ are specified for two patches. (b) and (c) show how we compute interior tangents and normals when none are prescribed. (b) n_1 and n_2 are the original normals of close petals P_1 and P_2 respectively. Their average n_c is the prescribed normal. (c) t_1 and t_2 are the original interior tangents of patch P_1 and P_2. t and $-t$ are their projections onto the direction of $t_2 - t_1$.

Tangent modification. The tangent modification is given by

$$p^{new} = p + s\left(\left(\lambda_1^r t_1 - a_1\right)x^1 + \left(\lambda_2^r t_2 - a_2\right)x^2\right)$$

where t_1 and t_2 are prescribed tangents at a point and r is the subdivision level. We scale the prescribed tangents to have the same length as a_1 and a_2 on the top level, to ensure that if t_i has the same direction as a_i, no modification ocurs on any level.

In the case of interior tangent interpolation along a singular edge, we apply this rule using the tangent to the singular curve as t_2 and interpolated interior tangent as t_1. s satisfies $0 < s \leq 1$; we use the value 3/4, but different value may work better depending on the mesh.

Normal modification. Normal modification is used for closed petals when a normal is prescribed. Using the same notation as before, to ensure that the surface has normal n at the vertex v we modify the vector of control points as follows:

$$p^{new} = p - s((a_1, n)nx^1 + (a_2, n)nx^2)$$

where s is a blending coefficient between 0 and 1; similarly, we use 3/4 in this case.

Note that the rules requiring prescribed normal or interior tangent are just elementary building blocks which can be used to define a surface; the fact that we use such rules does not mean that we will actually require a user to specify the tangent by hand, just as using Bezier curves as internal representation does not mean that independent Bezier points have to be manually prescribed for each segment. In most cases, reasonable interior tangents and normal data can be inferred from the mesh with minimum user input. We concider how this can be done in two typical situations.

Control of tangent direction along a curve. We summarize how our rules ensure that the surface has the prescribed tangent along a curve. Whenever a new vertex is inserted on a singular edge connecting two simple singular vertices with prescribed interior tangents, we assign it a tangent, making it into a new (simple) singular vertex with prescribed interior tangent. This tangent is used in tangent modification calculations.

It is not trivial to show that this subdivision process produces a surface with desired normals, and the resulting surfaces are C^1. The proof of C^1 continuity will be described in detail in a separate paper. Here we provide only an outline for such a proof (see Appendix B).

Tangent plane continuity across a singular curve. As we have mentioned the rules described above provide the least common denominator for a variety of ways to specify surface behavior along singular curves. We consider one of the most common scenarios: suppose two patches are required to share a tangent plane along a singular curve, but no normal or tangent is provided at the vertices of the curve.

In this case interior tangents can be inferred from the mesh. First, for each vertex on the curve, we compute its interior tangents in both patches using standard tangent plane rules. Then we specify the projection of each interior tangent onto their difference to be the specified interior tangent on both sides (Figure 6(c)). Models in Figures 10 and 11 were obtained in this way.

If tangent plane continuity is required for more than two patches user input appears to be necessary to specify a meaningful common tangent plane.

Shared normal at a singular vertex. Another common case is when it is necessary for several closed petals at a vertex to have a common normal direction. We can first compute the normal for each sector using the standard Loop subdivision rule, and then take the average of them as the specified common normal (Figure 6(b)). This approach was used in Figures 5(a) and 7.

We found these two cases to be most common but clearly it is possible to specify relations between patches in many different ways; in each case, an intermediate step should be used to convert the user-specified data to the form accepted by our algorithm.

Implementation. Most of the implementation is identical to implementation of piecewise smooth subdivision for manifold meshes. The most significant difference is the need to represent nonmanifold meshes. In our implementation, we use triangles as the main object class. Top-level triangles contain pointers to adjacent triangles and vertices; the triangles on finer levels have pointers only to vertices, descendant and parent triangles. For singular edges we use the following approach. Suppose triangles $T_1, T_2, \dots T_n$, $n > 1$ share an edge. Then T_i contains a pointer to $T_i + 1$ for $i < n$, and T_n contains a pointer to T_1. This is compatible with manifold case $n = 2$ (T_1 points to T_2 and T_1 points to T_2) and allows us to obtain all necessary information about local structure.

4 Results

We have implemented the algorithm described in the previous sections. The following images are examples of surfaces produced using our algorithms. The control mesh for the surfaces are rendered as gray line; for some models, the interior structures are shown in cross-sections.

In Figures 7 we specify the same normal directions for several manifold patches at a vertex. Notice that the surfaces are tangential near the fixed singular vertex. Figure 9 shows the case of dart petals. The patches on two sides of the singular edge also have the same tangent plane along the edge. Figure 10 shows the case when several patches have the same tangent plane along a singular curve. In Figure 11, we demonstrate how different surfaces are obtained by specifying different combinations of patches to share tangent plane along singular curve. We apply our algorithm to the X29 plane model. The original model is being modified to ensure the connectivity and preserve nonmanifold structure (Figure 12).

Our scheme was applied to create an approximate model of the human heart which has typical nonmanifold structures of biological models. Different muscle layers are abstracted as different manifold patches and these patches are connected along singular curves (Figure 13). This model is intended to be used in immersed boundary simulations, and is based on the model developed by C. Peskin and D. McQueen.

Immmersed boundary method requires surfaces to be represented by collections of infinitely thin fibers; in the case of the heart, it was found that using surface geodesics as fibers matches well the actual structure of muscle fibers (Figure 13(c)).

5 Conclusions and Future Work

We have demonstrated how subdivision can be applied to arbitrary nonmanifold meshes and tangent plane constraints can be imposed across the singular edges and at singular vertices of the resulting nonmanifold surface. We have also implemented the combination of this scheme with the piecewise smooth subdivision schemes of [1], which is straightforward, as the two schemes share majority of the rules. In fact, one can interpret the scheme of [1] as a special case of our scheme, when some of the regular mesh edges and vertices are marked as extraordinary. However, the combined scheme while not being much different, requires more effort to describe [14].

A few important types of surfaces are excluded by our rules: for example, surfaces with conic features; developing schemes to generate such surfaces is one of the possible directions for future research.

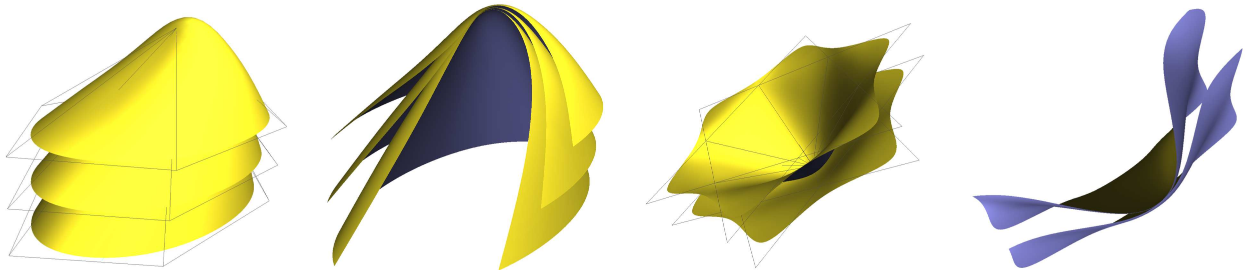
Due to curve and normal interpolation constraints the surface can exibit undulations near the places where the singular curves have complex shape. It is unlikely in such cases a significant improvement can be achieved in our setting while using fixed-coefficient subdivision rules with small support.

References

[1] Henning Biermann, Adi Levin, and Denis Zorin. Piecewise smooth subdivision surfaces with normal control. *Proceedings of SIGGRAPH 2000*, pages 113–120, July 2000. ISBN 1-58113-208-5.

[2] Jules Bloomenthal and Keith Ferguson. Polygonization of non-manifold implicit surfaces. *Proceedings of SIGGRAPH 95*, pages 309–316, August 1995. ISBN 0-201-84776-0. Held in Los Angeles, California.

[3] André P. Guéziec, Frank Bossen, Gabriel Taubin, and Cláudio T. Silva. Efficient compression of non-manifold polyg-



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

Figure 7: (a) Surface with three manifold patches interpolating the common point smoothly with the prescribed normal direction. (b) Cross-section of (a). (c) Surface with two patches interpolate common point smoothly. (d) Cross-section of (c).

onal meshes. *IEEE Visualization '99*, pages 73–80, October 1999.

[4] Andreas Hubeli and Marcus Gross. Fairing of non-manifold models. In *IEEE Visualization 2000*, Salt Lake City, Utah.

[5] A. Levin. Analysis of quazi-uniform subdivision schemes. Submitted, 1999.

[6] A. Levin. Combined subdivision schemes for the design of surfaces satisfying boundary conditions. *Computer Aided Geometric Design*, 16(5):345–354, 1999.

[7] A. Levin. Interpolating nets of curves by smooth subdivision surfaces. In *Proceedings of SIGGRAPH 99, Computer Graphics Proceedings, Annual Conference Series*, pages 57–64, 1999.

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

[9] Yi Luo and Gábor Lukács. A boundary representation for form features and non-manifold solid objects. *SMA '91: Proceedings of the First Symposium on Solid Modeling Foundations and CAD/CAM Applications*, pages 45–60, June 1991.

[10] Ahmad Nasri. Interpolating meshes of boundary intersecting curves by subdivision surfaces. *The Visual Computer Journal*, 16(1), 2000.

[11] Jovan Popovic and Hugues Hoppe. Progressive simplicial complexes. *Proceedings of SIGGRAPH 97*, pages 217–224, August 1997. ISBN 0-89791-896-7. Held in Los Angeles, California.

[12] Jarek Rossignac. Structured topological complexes: a feature-based api for non-manifold topologies. *SMA '97: Proceedings of the Fourth Symposium on Solid Modeling and Applications*, pages 1–9, May 1997.

[13] O. Volpin, A. Sheffer, M. Bercovier, and L. Joskowicz. Mesh simplification with smooth surface reconstruction. *Computer-aided Design*, 30(11):875–882, 1998.

[14] Lexing Ying and Denis Zorin. Nonmanifold subdivision. Technical Report, Computer Science Dept., New York Univ.

[15] Denis Zorin. A method for analysis of c^1-continuity of subdivision surfaces. *SIAM Journal of Numerical Analysis*, 37(4), 2000.

A Subdominant eigenvectors

Most of the subdominant eigenvectors required by the algorithm are the same as in the case of piecewise smooth manifold subdivision. The only new case is those for the disk rule.

Subdominant eigenvectors for disk rule. For a disk rule with degree k, its subdivision matrix is of size $(k+1)\times(k+1)$. In the following formulae, subscript c denotes the coefficient for center vertex and i is in the range $\{0, 1, \cdots, k-1\}$.

$$l_c^0 = 1, l_i^0 = 0$$

$$x_c^1 = 0, x_i^1 = \sin(2\pi i/k)$$
$$l_c^1 = 0, l_i^1 = 2/k\sin(2\pi i/k)$$
$$x_c^2 = 0, x_i^2 = \cos(2\pi i/k)$$
$$l_c^2 = 0, l_i^2 = 2/k\cos(2\pi i/k)$$

B Sketch of the proof of C^1-continuity

Analysis of the subdivision rules in almost all cases is identical to the analysis of the rules for piecewise smooth surfaces. The only case that requires special consideration is C^1 continuity of the surface at the singular curves with prescribed interior tangents along the whole curve. Complete analysis is rather tedious and will be described in a separate paper; while it is based on known techniques, the application of these techniques is nontrivial and requires introduction of extensive notation. Here we outline the basic steps of the verification.

As it is the case for all commonly used subdivision schemes, the analysis has two parts: regular case analysis and analysis of the behavior near extraordinary vertices.

Regular case. All new vertices inserted on the singular edges are regular (i.e. have three adjacent triangles in each petal). This means that the coefficients of the rules are fixed and the first subdivision step can be combined with second in an explicit form; thus, for vertices immediately adjacent to the boundary we get (see Figure 8 for notation)

$$p_2^{j+1} = (3+2s)\,p_0^j + (1+s)\,p_1^j + (3-2s)\,p_2^j + (1-2s)\,p_3^j + s p_4 + s/2t^j$$

The rules for vectors t are one dimensional rules, e.g. midpoint subdivision rules, with an additional factor 1/2: $t_0^{j+1} = t^j/2$; $t_1^{j+1} = (t_0^j + t_1^j)/4$. As we compute tangents for newly inserted vertices on the singular edge also using a simple subdivision scheme, we can regard the tangent as just another control.

In this formulation, the scheme reduces to a variation of a quasiuniform scheme [5], with additional tangent data added along the boundary. We have analyzed the scheme for the range of s from 0 to 1 and verified that it is C^1-continuous in all these cases.

To verify C^1-continuity we need to consider convergence of the divided difference scheme, i.e. the subdivision scheme $2S'$ which satisfies $S'\Delta p = \Delta S p$, where Δ is the vector of difference operators in directions $(1,0)$ and $(0,1)$. Analysis is based on the observation that convergence and continuity of a scheme follows from contractivity of the corresponding difference scheme, which in the case of quasiuniform schemes can be reduced to contractivity of a a uniform

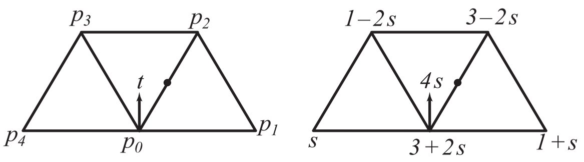


Figure 8: Mask of subdivision combined with tangent modification near a regular vertex on a singular edge, all coefficients are scaled by 1/8.

scheme and a one-dimensional matrix scheme [5]. Indeed, note that away from the boundary the only rules that are used are the standard quartic box spline subdivision rules. At any point away from the boundary, after sufficient number of subdivision steps, the limit can be computed using only the box spline rules. This means that we only need to prove existence and continuity of the limit functions at the boundary. We observe that all the irregular rules included in S' can be captured by a one-dimensional matrix subdivision scheme acting on the vectors $P_i = [[\Delta_1 p]_{0i}, [\Delta_1 p]_{1i}, [\Delta_2 p]_{0i}, [\Delta_2 p]_{1i}, \tilde{t}_i]$, where $[\Delta_j p]_{mi}$ is the forward difference in direction $(1,0)$ or $(0,1)$ at p_{mi} and $\tilde{t}_i = 2^s t_i$ for subdivision level s. The subdivision rules for this one-dimensional matrix scheme T' can be written as

$$P_{2i} = A_1^{\text{even}} p_{i-1} + A_0^{\text{even}} p_i + A_{-1}^{\text{even}} p_{i+1}$$
$$P_{2i+1} = A_2^{\text{odd}} p_{i-2} + A_1^{\text{odd}} p_{i-1} + A_0^{\text{odd}} p_i + A_{-1}^{\text{odd}} p_{i+1}$$

The matrix coefficients for the even rule are

$$A_{-1}^{\text{even}} = \begin{bmatrix} -4s+2 & 0 & 0 & 0 & 0 \\ 4s & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix},$$
$$A_0^{\text{even}} = \begin{bmatrix} 6-4s & 0 & -2s & 0 & 8s \\ 5+4s & 1 & 2s & 0 & -8s \\ 0 & 0 & 6 & 0 & 0 \\ 0 & 0 & 2 & 4 & 0 \\ 0 & 0 & 0 & 0 & 8 \end{bmatrix}, A_1^{\text{even}} = \begin{bmatrix} 0 & 0 & -2s & 0 & 0 \\ 1 & 0 & 2s & 0 & 0 \\ 0 & 0 & 2 & 0 & 0 \\ 0 & 0 & 2 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix}.$$

The matrix coefficients for the odd rule are

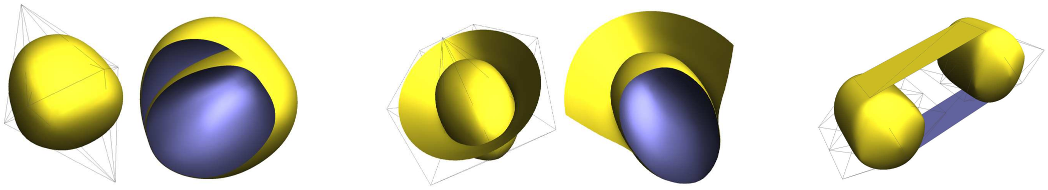
$$A_{-1}^{\text{odd}} = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ -4s & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix}, A_0^{\text{odd}} = \begin{bmatrix} 6-4s & 0 & 0 & 0 & 0 \\ 4s+2 & 2 & 0 & 0 & 0 \\ 0 & 0 & 2 & 0 & 0 \\ 0 & 0 & -2s & 2 & 8s \\ 0 & 0 & 0 & 0 & 4 \end{bmatrix},$$
$$A_1^{\text{odd}} = \begin{bmatrix} -4s+2 & 0 & -2s & 0 & 8s \\ 4s+4 & 0 & 2s & 0 & -8s \\ 0 & 0 & 6 & 0 & 0 \\ 4s & 0 & 4 & 2 & -8s \\ 0 & 0 & 0 & 0 & 4 \end{bmatrix}, A_2^{\text{odd}} = \begin{bmatrix} 0 & 0 & -2s & 0 & 0 \\ 0 & 0 & 2s & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 2s & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

Under some technical assumptions, a matrix scheme S converges and is continuous, if $\gamma = \left\| S'^N \right\|_\infty < 1$ for some N where S' is the difference scheme. Analysis of the difference scheme T'' for the matrix scheme T' shows that it is contractive: we found that $\left\| (T'')^7 \right\|_\infty < 1$ for $0 \le s \le 1$; together with some easily verifiable additional conditions, this is is known to prove that the scheme converges. We compute $\left\| T'' \right\|_\infty$ explicitly as a function of s using a symbolic algebra system.

Extraordinary vertices. If we regard interior tangents as additional control points,we can write the subdivision matrix M at extraordinary vertices, such that the vector of control points p (including additional control points for tangents) satisfies $Mp^j = p^{j+1}$. Using standard arguments, one can infer C^1-continuity of the scheme from regularity and injectivity of the corresponding characteristic map. Algorithms of [15] can be used to verify the properties of the characteristic map, as they require only estimates of the norm of the first and second difference schemes, which we have obtained to verify C^1 continuity in the regular case.

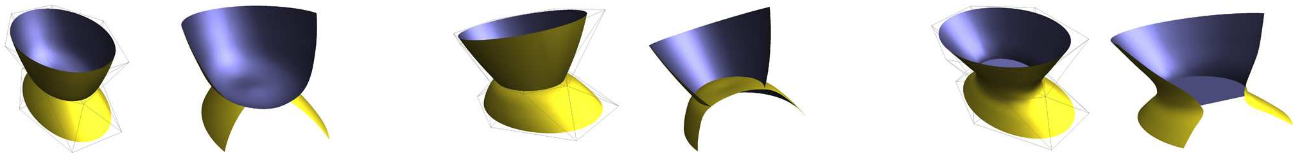
(a) (b)

Figure 9: (a) Surface with an attaching patch. Two patches share same tangent plane along singular curve and the endpoints of singular curve are dart nonmanifold vertex. (b) Two attaching cones. Four patches share same tangent plane along singular curve and the endpoints of cones are dart nonmanifold vertex.



(a) (b) (c)

Figure 10: Patches sharing some tangent plane along singular curve. (a) A ball with an interior patch. (b) A ball with an exterior attaching ring. (c) Two cylinders connected by two planes.



(a) (b) (c)

Figure 11: Three different ways to join three patches. In each case, two out of three patches share tangent planes along the common singular curve.

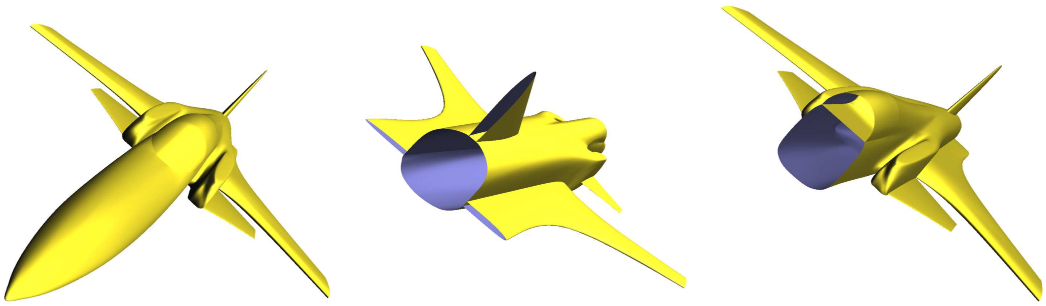
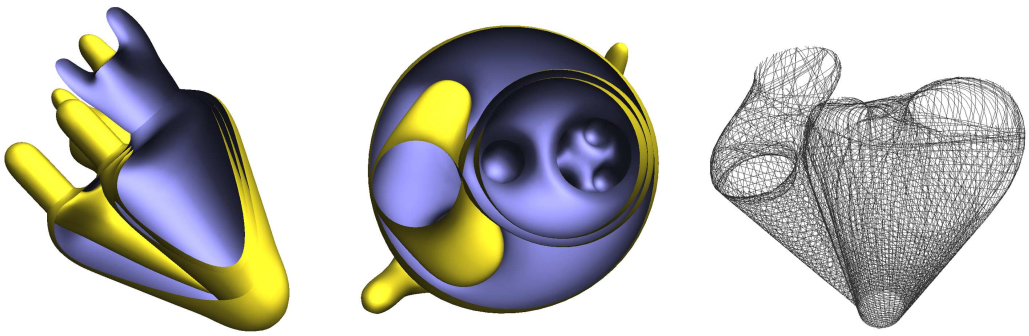


Figure 12: The X29 plane model. *Left:* The outlook of the plane. *Middle:* Nonmanifold structure where the wings attach the plane body. *Right:* Nonmanifold structure at the cockpit.



(a) (b) (c)

Figure 13: The human heart model. *Left and Middle:* Cross-sections from side and bottom. *Right:* Muscle fibers (geodesics) generated from the model.