Efficient, Fair Interpolation using Catmull-Clark Surfaces

Mark Halstead* Michael Kass Tony DeRose†
Apple Computer, Inc.

Abstract

We describe an efficient method for constructing a smooth surface that interpolates the vertices of a mesh of arbitrary topological type. Normal vectors can also be interpolated at an arbitrary subset of the vertices. The method improves on existing interpolation techniques in that it is fast, robust and general.

Our approach is to compute a control mesh whose Catmull-Clark subdivision surface interpolates the given data and minimizes a smoothness or "fairness" measure of the surface. Following Celniker and Gossard, the norm we use is based on a linear combination of thin-plate and membrane energies. Even though Catmull-Clark surfaces do not possess closed-form parametrizations, we show that the relevant properties of the surfaces can be computed efficiently and without approximation. In particular, we show that (1) simple, exact interpolation conditions can be derived, and (2) the fairness norm and its derivatives can be computed exactly, without resort to numerical integration.

CR Categories and Subject Descriptors: I.3.5 [Computer Graphics]: Computational Geometry and Object Modeling - curve, surface, solid, and object representations; J.6 [Computer-Aided Engineering]: Computer-Aided Design (CAD); G.1.2 [Approximation]: Spline Approximation.

Additional Key Words and Phrases: Computer-aided geometric design, B-spline surfaces, subdivision surfaces, thin-plate splines.

1 Introduction

The construction of smooth interpolating surfaces is becoming increasingly important in a number of applications including statistical data modeling, interactive design, and scientific visualization. Typical input to an interpolating method is a collection of points to be interpolated, and a "mesh" that describes the connectivity of the points. Normal vectors are sometimes also specified at some or all of the data points.

If the shape to be modeled is a deformed plane, techniques from function approximation, such as Clough-Tocher interpolation [5], can be used. An advantage of the Clough-Tocher interpolant is that the construction is local, meaning that modification of a data point affects only a local portion of the surface. However, a drawback of Clough-Tocher interpolation is that there are typically remaining degrees of freedom not directly constrained by the data. These extra degrees of freedom are often set using local heuristics and typically result in surfaces that are not "fair", that is, surfaces having extraneous bumps and wiggles. Another serious drawback to Clough-Tocher interpolation, and indeed to any method that requires continuity of parametric derivatives (so-called parametric continuity), is the inability to model surfaces of arbitrary topological type (cf. Herron [8]). It is not possible, for instance, to model a sphere or a deformed sphere using a Clough-Tocher interpolant.

Celniker and Gossard [3] recently presented an interpolation method that extends Clough-Tocher interpolation by setting the remaining degrees of freedom so as to minimize a fairness norm. The fairness norm they use is quadratic, so it can be minimized by solving a (sparse) linear system. As a result, their method is fast enough for interactive design. However, being based on Clough-Tocher interpolants, their technique is not capable of describing surfaces of arbitrary genus.

A number of interpolation methods appropriate for surfaces of arbitrary genus have been developed in recent years. A survey of these can be found in Lounsbery et al. [10]. The method developed by Shirman and Séquin [14] is a generalization of Clough-Tocher interpolation to surfaces of arbitrary topology. The generalization is achieved by replacing parametric continuity with first order geometric continuity (continuity of tangent planes). Like Clough-Tocher interpolation, Shirman-Séquin interpolants have degrees of freedom not directly constrained by the data, and local heuristics for setting these degrees of freedom have fallen well short of producing fair surfaces (see Figure 4).

Last year Moreton and Séquin [11] presented a method capable of producing fair interpolating surfaces of arbitrary genus. They achieved this in much the same way as Celniker and Gossard by solving a minimization problem using finite elements. However, rather than using Clough-Tocher elements and a quadratic fairness norm, Moreton and Séquin used biquintic Bézier patches and a fairness norm based on

*Work done while a summer intern from the University of California, Berkeley.

†Work done while on sabbatical leave from the University of Washington.

intrinsic measures of curvature variation. The surfaces produced are the most impressive to date, but improved shape and arbitrary genus are obtained at the expense of dramatically increased running time. It appears that Moreton and Séquin's method is far too expensive for use in an interactive environment today (computation time is on the order of hours). Another shortcoming of their method is that it constructs surfaces that are only approximately tangent plane smooth since inter-patch continuity is modeled using a penalty function added to the fairness norm. Finally, their surfaces are only curvature continuous within each biquintic patch.

Here we present a scheme that combines the speed of Celniker and Gossard's method with the ability to model tangent plane continuous surfaces of arbitrary genus. We do this by using a quadratic fairness norm similar to the one used by Celniker and Gossard together with Catmull-Clark subdivision surfaces. We show that Catmull-Clark surfaces offer a number of advantages over previous methods based on piecewise polynomial elements; these include:

- They are curvature continuous everywhere except at a finite number of isolated "extraordinary" points.
- The high order of continuity is obtained with very few control points, meaning that the dimension of the space over which the optimizer must search is far lower for Catmull-Clark surfaces than for the method described by Moreton and Séquin.
- They reduce to traditional bicubic B-splines when the points to be interpolated form a regular rectangular grid. It should therefore be possible to more smoothly incorporate them into existing geometric modeling systems.

The use of Catmull-Clark surfaces presents some challenges, however. First, Catmull-Clark surfaces do not generally interpolate their control points, so to achieve interpolation, a system of interpolation constraints must be solved. The constraints relate the data points and normals to be interpolated with points and normals on the final surface. Formulating the interpolation constraints at first appears problematic for a Catmull-Clark surface because the surface is defined as the limit of an infinite number of subdivisions. We show that it is possible to derive *closed form* expressions for these constraints. A second challenge posed by Catmull-Clark surfaces is that efficient surface optimization depends on fast and reliable evaluation of the fairness norm and its derivatives. We show that it is possible to evaluate the fairness integral and its derivatives *exactly*, without resort to numerical integration, even though Catmull-Clark surfaces do not possess a closed form polynomial representation.

Figure 5 illustrates the basic idea of our approach. The original mesh is shown in the upper left. Subdividing it using Catmull-Clark subdivision results in the surface shown in the lower left. The surface approximates, but does not interpolate the vertices of the original mesh. By solving the system of interpolation constraints, we obtain a new mesh which is shown in the upper center. Subdividing the new mesh results in the surface in the lower center which does interpolate the vertices of the original mesh. Unfortunately, the direct application of the interpolation conditions to the mesh causes undesirable undulations in the surface. To combat this difficulty, we subdivide the mesh to add new degrees of freedom, and we set these new degrees of freedom to minimize a fairness measure subject to the interpolation constraints. The resulting mesh is shown in the upper right of Figure 5 and the corresponding subdivision surface is shown in the lower right. Note that minimizing the fairness measure removes the spurious undulations introduced by the direct application of the interpolation constraints.

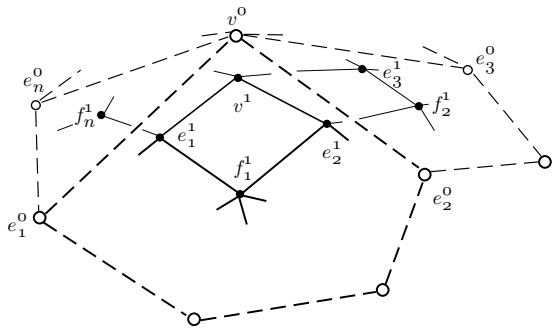

Figure 1: The situation around a vertex v^0 of order n.

The remainder of the paper is structured as follows. In Section 2 we provide some necessary background on subdivision surfaces in general, paying particular attention to Catmull-Clark surfaces. In Section 3, we derive the linear constraints on a Catmull-Clark mesh which guarantee that the surface interpolates given points and normals. We also show that applying these constraints directly to a mesh results in a surface which solves the interpolation conditions, but is unsatisfactory because of spurious wiggles. Then, in Section 4, we show how to reduce these artifacts by adding additional degrees of freedom through subdivision, and then setting them by optimizing a fairness norm based on the membrane/plate energy. Several implementation details along with performance statistics are provided in Section 5. In Section 6 we present a number of examples, and provide some comparisons to previous methods. Finally, in Section 7 we summarize our findings and describe several avenues of future research.

2 Subdivision Surfaces

In 1974 Chaikin [4] introduced the idea of generating a curve from a polygon by successively refining the polygon with the addition of new vertices and edges. In 1978, Catmull and Clark [2] and Doo and Sabin [6] generalized the idea to surfaces. In these schemes, an initial control mesh is refined by adding new vertices, faces and edges at each subdivision step. In the limit as the number of subdivision steps goes to infinity, the control mesh converges to a surface. With careful choice of the rules by which new vertices, edges and faces are introduced, it is possible to show that the limiting surface exists, is continuous, and possesses a continuous tangent plane. The Doo-Sabin subdivision rules generalize the subdivision rules for biquadratic B-splines, and the Catmull-Clark subdivision generalizes bicubic B-splines. An example of a Catmull-Clark surface of genus 3 is shown in Figure 3. A more recent method developed by Loop [9] generalizes quartic triangular B-splines. We focus on the Catmull-Clark scheme primarily because of the popularity of bicubic patches, however, much of the analysis we present is applicable to a wide class of subdivision schemes including those of Doo-Sabin and Loop.

When dealing with spline surfaces it is often helpful to maintain the distinction between global and local control meshes. By a local control mesh, we mean a subset of the

global mesh that influences a local region of the surface. Toward this end we use carets to denote global quantities.

Let $\widehat{M}^0$ denote the initial mesh, and let $\widehat{M}^i$ denote the mesh produced after i applications of the Catmull-Clark subdivision step. To describe the $i+1$-st subdivision step, consider the neighborhood of a vertex v^i of $\widehat{M}^i$ surrounded by n edge points $e_1^i, ..., e_n^i$ and n faces, as shown in Figure 1 for $i = 0$. Such a vertex is said to be of order n. As indicated in Figure 1, a new face point $f_1^{i+1}, ..., f_n^{i+1}$ is placed at the centroid of each face of $\widehat{M}^i$. Each new edge point $e_1^{i+1}, ..., e_n^{i+1}$ is then computed by taking an average of surrounding points. Specifically,

$$e_j^{i+1} = \frac{v^i + e_j^i + f_{j-1}^{i+1} + f_j^{i+1}}{4},$$

where subscripts are to be taken modulo n. Finally, a new vertex point v^{i+1} is computed as

$$v^{i+1} = \frac{n-2}{n} v^i + \frac{1}{n^2} \sum_j e_j^i + \frac{1}{n^2} \sum_j f_j^{i+1}.$$

The Catmull-Clark subdivision process is such that:

- The surfaces can be of arbitrary genus since the subdivision rules can be carried out on a mesh of arbitrary topological type.
- After the first subdivision step all faces are quadrilaterals.
- Except at extraordinary vertices (vertices of order $n \neq 4$) the limiting surface can be shown to converge to a bicubic B-spline. The surface is therefore curvature continuous except at extraordinary vertices.
- The number of extraordinary vertices is fixed, and is equal to the number of extraordinary vertices in $\widehat{M}^1$, the mesh produced after the first subdivision step.
- Near an extraordinary vertex the surface does not possess a closed form parametrization; it consists of an infinite number of bicubic patches that converge to a limit point. The surface can be shown to have a well defined tangent plane at the limit point, but the curvature there is generally not well defined [1].

3 Interpolation using Subdivision Surfaces

Given a mesh $\widehat{I}$ of arbitrary topological type, the idea is to generate a control mesh $\widehat{M}^0$ such that the subdivision surface it defines interpolates some or all of the vertices of $\widehat{I}$. It is also possible to constrain the surface to have a specified normal at each interpolation point.

Nasri [12] generates interpolating surfaces using the biquadratic formulation of Doo and Sabin [6]. Like biquadratic B-splines, Doo-Sabin surfaces interpolate the centroid of each face in the control mesh. Thus a linear constraint on the control vertices can be generated for each interpolation point and the resultant system solved for the desired control mesh[1]. It appears that Nasri had no simple formulation for the surface normal at the centroid, and so was unable to specify normals at these points.

[1]Although Nasri does not mention it, it is possible for the coefficient matrix in the linear system to be singular.

To generate interpolating surfaces for other subdivision schemes we need a method of determining the position and normal at a set of points on the limit surface. Because the surface is the result of repeated application of a subdivision step, we can analyze the behavior of a small neighborhood of points as they converge to the limit surface in order to determine the surface properties at the point of convergence.

3.1 Interpolation Conditions

After one subdivision step there arises an arrangement of vertices that persists (i.e. the same topology will be observable) for any number of subsequent subdivisions. To analyze the limiting behavior of the surface near a vertex it is therefore convenient to introduce a matrix that describes the subdivision process locally, that is, in the neighborhood of the vertex [6]. It is not necessary to compute local subdivision matrices in practice; they are simply tools used to derive formulas describing the limiting behavior of the surface.

Let v^i be a vertex of order n of the mesh $\widehat{M}^i$, let $V_n^i = (v^i, e_1^i, ..., e_n^i, f_1^i, ..., f_n^i)^T$ be the column vector of vertices in the neighborhood of v^i, and let V_n^{i+1} be the corresponding column vector of points in the neighborhood after subdivision. Since the points in V_n^{i+1} are computed by linear combinations of the points in V_n^i, we can use a square matrix $\mathbf{S}_n$ to express the subdivision:

$$V_n^{i+1} = \mathbf{S}_n V_n^i.$$

For instance, for Catmull-Clark surfaces the matrix $\mathbf{S}_4$ is

$$\mathbf{S}_4 = \frac{1}{16} * \begin{pmatrix} 9 & \frac{3}{2} & \frac{3}{2} & \frac{3}{2} & \frac{3}{2} & \frac{1}{4} & \frac{1}{4} & \frac{1}{4} & \frac{1}{4} \\ 6 & 6 & 1 & 0 & 1 & 1 & 0 & 0 & 1 \\ 6 & 1 & 6 & 1 & 0 & 1 & 1 & 0 & 0 \\ 6 & 0 & 1 & 6 & 1 & 0 & 1 & 1 & 0 \\ 6 & 1 & 0 & 1 & 6 & 0 & 0 & 1 & 1 \\ 4 & 4 & 4 & 0 & 0 & 4 & 0 & 0 & 0 \\ 4 & 0 & 4 & 4 & 0 & 0 & 4 & 0 & 0 \\ 4 & 0 & 0 & 4 & 4 & 0 & 0 & 4 & 0 \\ 4 & 4 & 0 & 0 & 4 & 0 & 0 & 0 & 4 \end{pmatrix}.$$

Repeated subdivision is expressed by repeated multiplication and hence powers of $\mathbf{S}_n$, so

$$V_n^{i+1} = \mathbf{S}_n^i V_n^1.$$

The properties of the limit surface will be governed by the properties of V_n^{i+1} as i approaches infinity. Since V_n^{i+1} is the image of V_n^1 under $\mathbf{S}_n^i$, the eigenstructure of $\mathbf{S}_n$ naturally plays a key role.

In Appendix A we analyze the behavior of the limit surface in terms of the matrix $\mathbf{S}_n$ by building on the analytical techniques of Doo and Sabin [6] and Ball and Storry [1]. Like Loop [9], we find that the positions and normals of the limit surface can be expressed explicitly in terms of the vertices of the control mesh. However, whereas Loop's analysis was peculiar to his subdivision surfaces, our analysis applies to any subdivision scheme whose local matrix $\mathbf{S}_n$ satisfies the conditions listed in Appendix A. In particular, our analysis exposes the following simple dependence between the left eigenvectors of $\mathbf{S}_n$ and limit points and normals.

Let $\lambda_1 \geq \lambda_2 \geq \lambda_3$ be the three largest eigenvalues of $\mathbf{S}_n$ and let l_1, l_2, l_3 be the corresponding left eigenvectors. In Appendix A we show that a point v^1 having a neighborhood V_n^1 converges to the point

$$v^\infty = l_1 \cdot V_n^1 \tag{1}$$

and the normal vector to the surface at v^∞ is given by

$$N^\infty = c_2 \times c_3 \tag{2}$$

where $c_2 = l_2 \cdot V_n^1$ and $c_3 = l_3 \cdot V_n^1$, and where "$\times$" denotes vector cross product. Explicit formulas for l_1, l_2 and l_3 for Catmull-Clark surfaces can be found in Appendix A.

Equation 1 provides an interpolation condition that is linear in the control points of V_n^1, but Equation 2 at first appears to impose a quadratic constraint on V_n^1's control points. Fortunately, we can require a surface to have a given normal vector N, using the following two linear constraints:

$$N \cdot c_2 = 0 \qquad \text{and} \qquad N \cdot c_3 = 0 \tag{3}$$

In addition to providing interpolation constraints, the limit point and normal vector formulas can also be used to compute exact points and normal vectors on the surface for use during rendering [9]. The color images (Figures 3 through 7) have all been computed this way.

3.2 Solving the Interpolation Problem

Ignoring the interpolation of normals for the time being, we can use the interpolation condition in Equation 1 to compute a control mesh $\widehat{M}^0$ with the property that the subdivision surface it defines interpolates the vertices of a given mesh $\widehat{I}$. It is natural to do this by selecting $\widehat{M}^0$ to have the same mesh topology as $\widehat{I}$, that is, the same number and connectivity of vertices, faces, and edges. This approach leads to a square linear system of the form

$$\mathbf{A}x = b \tag{4}$$

where x is the column vector of the unknown vertex coordinates in $\widehat{M}^0$, and b is the corresponding column vector of vertex coordinates of $\widehat{I}$. The rows of the square matrix $\mathbf{A}$ are determined by the interpolation conditions and mesh topology. In some cases, the matrix $\mathbf{A}$ is singular, so we use a least-squares solution to Equation 4. An example is shown in Figure 5. The original mesh is shown in the upper left. Subdividing it according to the usual Catmull-Clark rules results in the lower-left surface which approximates, but does not interpolate the vertices of the original mesh. By solving Equation 4, we obtain a new mesh which is shown in the upper center. Subdividing the new mesh according to the usual Catmull-Clark rules gives the surface in the lower center which does interpolate the vertices of the original mesh.

4 Fairing

The surface in the lower center of Figure 5 is curvature continuous almost everywhere and interpolates the vertices of the original mesh. Nonetheless, for many purposes it is an unsatisfactory interpolating surface because of its excessive undulations. These undulations appear to be artifacts of the interpolation process since they are not indicated by the shape of the original mesh. For example, the surface has a number of concavities where the original mesh is convex. Note that some of the undulations are present in the ordinary approximating Catmull-Clark surface, but they have become more severe and objectionable in the interpolating surface. This difference is typical of interpolating and approximating surfaces.

Nothing in our formulation of the interpolation conditions in Section 3 prohibits or discourages undulations in the surface, so this type of behaviour should not be surprising. In order to improve the quality of the interpolant, we introduce additional degrees of freedom into the surface by subdivision, and then set the degrees of freedom by optimizing a fairness norm on the surface subject to a set of linear constraints given by the interpolation conditions.

4.1 Evaluating the Fairness Norm

Celniker and Gossard [3] were able to improve the quality of interpolating surfaces using a fairness norm based on a linear combination of the energy of a membrane and a thin plate. Without any fundamental changes, the norm can be given directional preferences and nonuniform weighting over the surface, but for clarity of presentation, we consider the isotropic uniform case:

$$E(W) = \alpha E_m(W) + \beta E_p(W) \tag{5}$$

where $E_m(W)$ and $E_p(W)$ denote the membrane and thin-plate energies respectively:

$$\begin{aligned} E_m(W) &= \int\int \|W_u\|^2 + \|W_v\|^2 \, du \, dv \\ E_p(W) &= \int\int \|W_{uu}\|^2 + 2\|W_{uv}\|^2 + \|W_{vv}\|^2 \, du \, dv, \end{aligned}$$

and where $W(u,v) = (x(u,v), y(u,v), z(u,v))$ is a parametric representation of the surface, where subscripts on W represent parametric derivatives, and where α and β are freely selectable weights.

Since the membrane/plate norm is defined in terms of a parametric representation of the surface, it cannot be directly applied to Catmull-Clark surfaces since in general they have no "natural" parametrization near extraordinary points. The remainder of this section describes how we extend the definition of the norm in a way that can be used with Catmull-Clark surfaces. As we show below, the extended norm will be constructed to be quadratic in the control points of the mesh. The optimization can consequently be performed quickly without iteration by solving a linear system. Moreover, there is a unique minimum since the Hessian of the norm is symmetric and positive definite.

The membrane/plate norm can be evaluated without modification on a bicubic patch W as follows. First, we note that the norm can be written as $E = E_x + E_y + E_z$, where E_x depends only on the x component of W, E_y only on the y component and E_z only on the z component of W. Let P_x be a 16-element column vector of positions of the x coordinates of the control points W. Figure 2(a) schematically depicts a 16 element control net and the bicubic patch it defines. The x component of the fairness norm for the patch can be expressed as

$$E_x = P_x^T \cdot \mathbf{K} \cdot P_x \tag{6}$$

where the entries of the 16×16 matrix $\mathbf{K}$ can be computed exactly from the integrals in Equation 5 for bicubic B-spline basis functions. Similar formulas hold for the y and z components.

Figure 2(b) depicts a mesh that includes an extraordinary point. The region of the limit surface corresponding to the central face in the mesh is shown at the center bottom, but the limit surface is not in general a parametric polynomial, so we cannot directly apply the membrane/plate norm used above for a bicubic mesh. However, we can subdivide the

mesh in Figure 2(b) to obtain the mesh in (c). After subdivision, the limit surface is divided into four subpatches. Three of these subpatches (shown shaded in (c)) are bicubic B-splines, so on these patches we can in principle evaluate the fairness norm exactly. By repeating this procedure we can write an infinite series for the fairness norm of the original extraordinary patch of Figure 2(b). In order to fully define the series, we must choose a parametrization for each of the B-spline subpatches during subdivision. Unfortunately, the most straightforward way to assign the parametrizations causes the infinite series for the thin plate energy to diverge (see Appendix B).

There are several methods that could be applied to overcome the problem of the divergent series. For instance, we might try to find an alternate method of parametrizing the subpatches that leads to convergent sequences. We are currently investigating this possibility, but we have found that the following method gives good results. Intuitively, we intend to modify the thin plate energy so that it integrates to zero for surface patches defined by planar and "regular" control meshes. For a bicubic mesh it is relatively clear that a regular mesh is one that is an affine image of Figure 2(a) since such a mesh has vanishing second derivatives. As shown in Appendix B, it is possible to generalize the notion of regularity for meshes containing an extraordinary vertex. It is also possible to measure the deviation of an arbitrary mesh of control points P from it's regular component P'. We therefore define the modified thin plate energy of P to be the thin plate energy of $P - P'$. In symbols, the norm we use can be written as

$$E(P) = \alpha E_m(P) + \beta E_p(P - P'). \tag{7}$$

We have written this norm as a function of the control mesh P rather than the limit surface that P defines. This is to emphasize that the norm is not, strictly speaking, a property of the limit surface. It is more appropriate to think of Equation 7 as a norm on meshes, because it is not generally the case that $E(P^i) = E(P^{i+1})$ where P^i and P^{i+1} denote the mesh after i and $i+1$ subdivisions. Although this might be considered a theoretical deficiency, it has posed no difficulties in practice.

Using the modified norm, the infinite series is a convergent geometric series, so we can express its limiting value analytically. Appendix B contains the relevant details, but the result is that we can exactly compute the entries of a new quadratic form $\mathbf{K}_n$ that can be applied around an extraordinary vertex of order n.

Now that we have defined the local fairness norm for patches surrounding extraordinary patches, we define the global fairness norm as the sum of the fairness norms over each of the patches using the standard membrane/plate norm for bicubic patches and the modified norm of Equation 7 for extraordinary patches. We can write the global fairness norm as $\widehat{P}^T\widehat{\mathbf{K}}\widehat{P}$ where $\widehat{\mathbf{K}}$ is a sparse matrix obtained from the various $\mathbf{K}_n$ by iterating over the individual vertices and collecting the entries into a global system, and where $\widehat{P}$ is a column vector containing the x, y and z coordinates of the control vertices in the global mesh $\widehat{M}^0$.

4.2 Minimizing the Fairness Norm

Since we have a global expression for the fairness norm, we are now in a position to express and solve the minimization problem. Given a mesh $\widehat{I}$ with t vertices, r of which are constrained to have a specified limit point and s of which are constrained to have specified normals, we seek the vector of $3t$ vertex coordinates $\widehat{P}$ such that the limit surface satisfies the $3r + 2s$ linear interpolation conditions and the fairness norm $\widehat{P}^T\widehat{\mathbf{K}}\widehat{P}$ is minimized over all possible $\widehat{P}$.[2] Because the constraints are linear and the norm is quadratic in the unknowns, this problem can be solved directly without iteration.

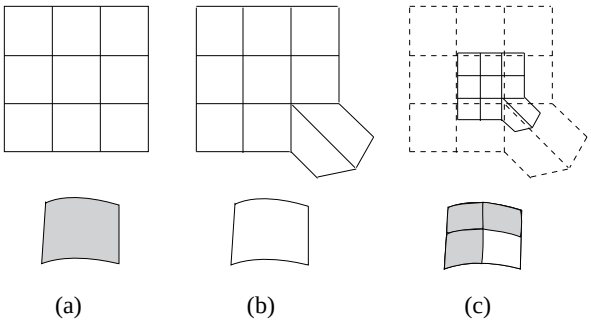

Figure 2: (a) A regular control mesh (above) which generates a bicubic B-spline patch on the limit surface (below). (b) A control mesh with an extraordinary point (above), and the extraordinary surface patch it defines (below). (c) The control mesh after one subdivision (above), and the four subpatches after subdivision (below). The three bicubic subpatches are shaded gray, and the remaining extraordinary subpatch is shaded white.

If we have only positional constraints, the x, y and z components of the mesh are independent, so the whole problem decouples into three completely independent optimizations, one for each component of the mesh. If normal vectors are to be interpolated, the x, y and z components of the mesh are no longer independent, so the problem must be solved as a single optimization. Even so, the x, y and z components of the mesh remain nearly decoupled (in the sense that the linear system is block diagonal except for a few off-diagonal terms) and sparse matrix methods exist that can exploit this fact[7].

The r position constraints and s normal constraints on the t mesh points can be represented by the equation $\mathbf{B}\widehat{P} = D$ where $\mathbf{B}$ is a $(3r+2s)\times 3t$ matrix and D is a vector of length $3r+2s$. Let $\mathbf{C}$ be the $3t$ by l matrix whose columns span the null space of $\mathbf{B}$ and let $\widehat{P}_0$ be any vector satisfying $\mathbf{B}\widehat{P}_0 = D$. Then all $\widehat{P}$ which satisfy the interpolation constraints can be written in the form $\widehat{P}_0 + \mathbf{C}R$ for some l-vector R. Therefore we wish to find the vector R that minimizes:

$$(\widehat{P}_0 + \mathbf{C}R)^T\ \widehat{\mathbf{K}}\ (\widehat{P}_0 + \mathbf{C}R) = R^T\mathbf{C}^T\widehat{\mathbf{K}}\mathbf{C}R + 2R^T\mathbf{C}^T\widehat{\mathbf{K}}\widehat{P}_0 + \widehat{P}_0^T\widehat{\mathbf{K}}\widehat{P}_0.$$

$\widehat{\mathbf{K}}$ is symmetric and positive definite, so R is found by setting the gradient of this function to zero:

$$\mathbf{C}^T\widehat{\mathbf{K}}\mathbf{C}R + \mathbf{C}^T\widehat{\mathbf{K}}\widehat{P}_0 = 0. \tag{8}$$

[2]Each of the t vertices has three coordinates, so the total number of unknowns is $3t$. Each position interpolation constraint imposes three conditions, one per coordinate, and each normal vector constraint imposes the two conditions in Equation 3.

5 Implementation

For simplicity and speed, our current implementation of the fairing process uses only positional constraints and exploits the fact that the linear systems for x, y and z decouple in this case. As a result, the implementation is able to compute the minimum energy mesh by solving three linear systems, each involving one third as many variables as Equation 8. To further speed the computation, each of these systems is solved using sparse-matrix methods.

Given a mesh $\widehat{I}$ whose vertices are to be interpolated, we must first choose the structure of the mesh $\widehat{M}^0$ whose vertices we compute. Our current implementation chooses $\widehat{M}^0$ to have the structure that would result from subdividing $\widehat{I}$ twice. This choice has two benefits. First, it adds enough extra degrees of freedom for the fairing to be effective. Second, it places enough new vertices between the interpolation points to ensure that the interpolation conditions for all vertices of $\widehat{I}$ are independent, making the construction of a sparse representation of the required null space easy.

Since we are considering a single component x, y or z at a time and not allowing normal constraints, we can still write the interpolation conditions as $\mathbf{B}\widehat{P} = D$ but now $\mathbf{B}$ is an $r \times t$ matrix and D is a vector of length r. We compute a sparse set of null-space vectors for $\mathbf{B}$ as follows. Suppose the ith row of $\mathbf{B}$ has k non-zero entries in columns $(a_1, a_2, \ldots, a_k)$. Because of the way the positional constraints decouple after two subdivisions, all other entries of $\mathbf{B}$ in those k columns are zero. As a result, it is an easy matter to find $k-1$ independent null-space column vectors which are zero except in rows $(a_1, a_2, \ldots, a_k)$. Collecting these for each row of $\mathbf{B}$ yields a collection of sparse vectors that completely span the null space of $\mathbf{B}$ unless $\mathbf{B}$ contains zero columns. If $(b_1, b_2, \ldots, b_m)$ are the zero columns of $\mathbf{B}$, we complete the null space by adding the m vectors Q_s, $1 \le s \le m$ where Q_s is one in the b_sth entry and zero elsewhere.

In addition to the null space, we need a feasible mesh $\widehat{P}_0$ which satisfies the constraints. We construct this mesh as follows. For each row i in $\mathbf{B}$, with non-zero entries in columns $(a_1, a_2, \ldots, a_k)$, set the entries of $\widehat{P}_0$ at indices $(a_1, a_2, \ldots, a_k)$ to D_i and set any remaining entries of $\widehat{P}_0$ to zero. Then since all the rows of $\mathbf{B}$ sum to one, the resulting $\widehat{P}_0$ will solve the equation $\mathbf{B}\widehat{P}_0 = D$.

Finally, given the null space basis $\mathbf{C}$ and the feasible mesh $\widehat{P}_0$, we compute the minimum energy mesh by solving Equation 8 three times using sparse LU decomposition, once for each component of the mesh. If the mesh is a regular square grid, the bandwidth of the linear system will be $O(\sqrt{n})$, and the linear system will take $O(n^2)$ time to solve. The running time is more difficult to analyze for general meshes, but the times we have observed to date are consistent with $O(n^2)$ performance.

6 Results

Figure 5 shows the complete process of interpolation and fairing. The original mesh is shown in the top left. The interpolating mesh is shown at top center. The faired, interpolating mesh is shown at top right. Below each mesh is the corresponding Catmull-Clark limit surface. Note that the spurious undulations in the interpolating limit surface are greatly reduced in the faired interpolating surface. The additional subdivisions in the faired interpolating mesh provide the degrees of freedom necessary to do this. For the examples presented in this paper, we set $\alpha = 0$ and $\beta = 1$.

Often it is desirable to fair only a local region of the surface, either to have more control over the fairing or because the number of vertices in the control mesh is large. In this case we select a subset of control vertices that are free to move and compute the solution to the constrained minimization over the surface patches affected by this set. Figure 6 illustrates this process. The user has selected a subset of 52 vertices that are allowed to vary during the minimization process. These vertices are highlighted in red. Other nearby vertices which influence the minimization, but are not allowed to change, are shown in magenta. After fairing, the undulations in the faired region have been reduced, but they persist in the unfaired regions. In this case, the fairing took .18 seconds on an SGI Crimson workstation.

Lounsbery et al. [10] have done a survey of the previously published interpolation methods and found that existing local interpolation schemes do an unsatisfactory job of constructing fair surfaces, even for the simple cases such a data sampled from a torus. To facilitate comparison with these methods, we have run our algorithm and a representative local interpolant, that of Shirman and Séquin [14], on the same coarsely sampled toroidal data set. The results are shown in Figure 4. The upper left shows the original mesh used as input for the interpolants. The upper right shows the surface produced by the Shirman-Séquin algorithm. The odd looking specular highlights in the Shirman-Séquin interpolant point out some interpolation artifacts which are typical of local methods. Global methods tend to have a different appearance. The surface in the lower left of Figure 4 is a Catmull-Clark surface that interpolates the original mesh using the methods of Section 3. This surface has different (lower frequency) artifacts than the Shirman-Séquin interpolant, but they are nonetheless objectionable. The surface in the lower right is an interpolating faired surface computed using our method. The surface has no visible artifacts, an observation confirmed by examining the surface from other viewpoints. The implementation took 36.5 seconds to fair the entire 600 point mesh at once on an SGI Crimson workstation.

The result of applying the interpolation algorithm to a more complicated model is shown in Figure 7. The original mesh is shown at the far left. The left center shows the ordinary approximating Catmull-Clark surface. Note the artifacts throughout the stem and where the stem meets the base. These artifacts are accentuated in the interpolating Catmull-Clark surface shown in the right center. In addition, the interpolating surface shows severe overshoot at the bottom of the stem. This type of overshoot is typical of interpolation without fairing. The far right shows the faired interpolating Catmull-Clark surface computed using our method. The artifacts along the stem and where the stem joins the base have been removed. Fairing the 1273 point mesh took 127.8 seconds on an SGI Crimson workstation.

7 Conclusions

We have described an efficient method for constructing fair surfaces that interpolate the vertices of a mesh of arbitrary topological type; normal vectors can also be interpolated at an arbitrary subset of the vertices. Our approach is to compute a control mesh describing a Catmull-Clark surface that interpolates the given data and minimizes a quadratic

norm that combines thin plate and membrane energies.

Our method improves on previous techniques by combining many of the strengths of the methods described by Celniker and Gossard and by Moreton and Séquin. Like Celniker and Gossard, we use a quadratic norm to achieve practical fairing at interactive rates. Like Moreton and Séquin, we use a representation capable of modeling arbitrary topological surfaces. In addition, the Catmull-Clark representation we use provides improved surface continuity with remarkably few degrees of freedom. More specifically, Celniker-Gossard surfaces meet with only tangent plane continuity along patch boundaries, and those of Moreton-Séquin meet with only approximate tangent plane continuity. Our surfaces, in contrast, are curvature continuous everywhere except at a finite number of isolated points.

Our work also provides two new analytical tools for analyzing and manipulating subdivision surfaces: limit point and normal vector analysis based on left eigenvectors of the local subdivision matrix, and a method for developing exact formulas for evaluating quadratic membrane/plate functionals and their derivatives.

As a topic for future research, we plan to investigate using the surfaces produced by our method as a starting point for minimizing the intrinsic "MVS" norm developed by Moreton and Séquin. We are also interested in developing subdivision schemes that are curvature continous everywhere.

References

[1] A. A. Ball and J. T. Storry. Conditions for tangent plane continuity over recursively defined B-spline surfaces. *ACM Transactions on Graphics*, 7(2):83–102, April 1988.

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

[3] George Celniker and Dave Gossard. Deformable curve and surface finite elements for free-form shape design. In *Proceedings of SIGGRAPH '91*, pages 257–265, July 1991.

[4] G. Chaikin. An algorithm for high speed curve generation. *Computer Graphics and Image Processing*, 3:346–349, 1974.

[5] R. Clough and J. Tocher. Finite element stiffness matrices for analysis of plate bending. In *Matrix Methods in Structural Mechanics (Proceedings of the conference held at Wright-Patterson Air Force Base, Ohio, 26-28 October 1965)*, pages 515–545, 1966.

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

[7] Gene H. Golub and Charles F. Van Loan. *Matrix Computations*. The Johns Hopkins University Press, Baltimore, 2nd edition, 1989.

[8] G. Herron. Techniques for visual continuity. In G. Farin, editor, *Geometric Modeling*, pages 163–174. SIAM, 1987.

[9] Charles T. Loop. Smooth subdivision surfaces based on triangles. M.S. Thesis, Department of Mathematics, University of Utah, August 1987.

[10] Michael Lounsbery, Stephen Mann, and Tony DeRose. Parametric surface interpolation. *IEEE Computer Graphics and Applications*, 12(5):45–52, September 1992.

[11] Henry P. Moreton and Carlo Séquin. Functional optimization for fair surface design. In *Proceedings of SIGGRAPH '92*, pages 167–176, July 1992.

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

[13] Malcolm Sabin. Recursive division singular points. Unpublished manuscript, June 1992.

[14] L. Shirman and C. Séquin. Local surface interpolation with Bézier patches. *Computer Aided Geometric Design*, 4(4):279–296, 1988.

Appendix

A Properties of the Limit Surface

To develop formulas for limit points and normals on subdivision surfaces, we examine the eigenstructure of the local subdivision matrix $\mathbf{S}_n$ associated with the subdivision scheme. (Some of the following analysis appears to have been developed independently by Sabin [13].)

Let $m = 2n + 1$ denote the size of $\mathbf{S}_n$, and let $\lambda_1 \geq \lambda_2 \geq ... \geq \lambda_m$ denote the eigenvalues of $\mathbf{S}_n$ with corresponding right eigenvectors $r_1, ..., r_m$ and left eigenvectors $l_1, ..., l_m$. If $\mathbf{S}_n$ is not defective, the right eigenvectors form a basis, and the left eigenvectors can be chosen so that (cf. Golub and Van Loan [7])

$$l_k \cdot r_j = \delta_{kj}. \tag{9}$$

Thus, assuming that $\mathbf{S}_n$ is not defective, the neighborhood V_n^1 can be expanded uniquely as

$$V_n^1 = c_1 r_1 + \cdots + c_m r_m \tag{10}$$

where the c's are geometric position vectors and where the r's are column vectors of scalars. The c_k, $k = 1, ..., m$ can be determined by dotting both sides of Equation 10 with l_k and using Equation 9:

$$l_k \cdot V_n^1 = c_1 l_k \cdot r_1 + \cdots + c_k l_k \cdot r_k + \cdots + c_m l_k \cdot r_m = c_k. \tag{11}$$

Using this expansion of V_n^1,

$$V_n^i = \mathbf{S}_n^i V_n^1 = \lambda_1^i c_1 r_1 + \cdots + \lambda_m^i c_m r_m.$$

For a non-trivial limit to exist as $i \to \infty$, it is necessary for the magnitude of the largest eigenvalue λ_1 to be 1. In this case,

$$V_n^\infty := \lim_{i \to \infty} V_n^i = c_1 r_1 = \left(l_1 \cdot V_n^1\right) r_1$$

For a subdvision scheme to be affine invariant (that is, independent of the coordinate system in which the calculation is performed), the points of $\widehat{M}^{i+1}$ must be affine combinations of the points in $\widehat{M}^i$, meaning that each of the rows of $\mathbf{S}_n$ must sum to one. In matrix form:

$$\mathbf{S}_n (1, ..., 1)^T = (1, ..., 1)^T.$$

In other words, the column vector of 1's is the eigenvector r_1 associated with eigenvalue 1. Since r_1 is a column vector of 1's, every point in the neighborhood converges to the point

$$c_1 = l_1 \cdot V_n^1 \tag{12}$$

on the limit surface. Stated more formally, we have proven that:

Proposition 1: A point v^1 of $\widehat{M}^1$ with neighborhood V_n^1 and local subdivision matrix $\mathbf{S}_n$, converges to the point

$$v^\infty = l_1 \cdot V_n^1$$

on the limit surface where l_1 is the left eigenvector of S_n associated with eigenvalue 1, assuming that S_n satisfies the following conditions:

i) $\mathbf{S}_n$ is not defective.
ii) $\mathbf{S}_n$ describes an affine invariant process.
iii) The magnitude of the largest eigenvalue is 1 and it has multiplicity 1.

Using a discrete Fourier analysis similar to the one described by Ball and Storry [1], one can show that for Catmull-Clark surfaces the above conditions on $\mathbf{S}_n$ hold and that

$$l_1 = \frac{1}{n(n+5)}(n^2, 4, ..., 4, 1, ..., 1),$$

meaning that

$$v^\infty = \frac{n^2 v^1 + 4\sum_j e_j^1 + \sum_j f_j^1}{n(n+5)}. \tag{13}$$

Equation 13 can be used as an interpolation condition on the points of $\widehat{M}^1$ by setting v^∞ to a point to be interpolated. Note that the interpolation conditions are on the vertices of $\widehat{M}^1$, not on the vertices of the initial control mesh $\widehat{M}^0$, since the analysis above requires that each face has exactly four edges. This apparent restriction poses no problem in practice since fairing requires the extra degrees of freedom present in $\widehat{M}^1$.

To develop an interpolation condition on normal vectors, we must determine the normal vector (if it exists) to the limit surface at v^∞. This normal vector can be simply computed from the eigenstructure of $\mathbf{S}_n$, as indicated by the following proposition.

Proposition 2: The normal vector to a subdivision surface at a limit point v^∞ corresponding to a vertex v^1 whose neighborhood is $\widehat{M}_n^1$ is the vector

$$N^\infty = c_2 \times c_3$$

where $c_2 = l_2 \cdot \widehat{M}_n^1$ and $c_3 = l_3 \cdot \widehat{M}_n^1$, assuming that the local subdivision matrix $\mathbf{S}_n$ satisfies the conditions of Proposition 1 in addition to:

iv) The eigenvalues $\lambda_1 = 1 \geq \lambda_2 ...$ are such that $\lambda_2 = \lambda_3 > \lambda_4$.

Proof sketch: The general idea behind the proof is to show that there is a common plane to which all points in the neighborhood are converging. The vector N^∞ will then be chosen to be perpendicular to this plane. Let u_j^i denote the vector from v^∞ to the j-th point p_j^i of the neighborhood $\widehat{M}^i$. Roughly speaking, if a common plane exists, then it should be possible to find an expression for a vector N^∞ that is perpendicular to each of the u_j^i's in the limit $i \to \infty$. Stated as an equation, we might seek a vector N^∞ such that

$$N^\infty \cdot u_j^i \to 0$$

for $j = 2, ..., m$ as $i \to \infty$. This does not quite work, however, because each u_j^i is approaching the zero vector, implying that the above condition would trivially hold for any vector N^∞. This problem is overcome by considering the unit vectors $\hat{u}_j^i$. Thus, we seek a vector N^∞ such that

$$N^\infty \cdot \hat{u}_j^i \to 0$$

for $j = 2, ..., m$ as $i \to \infty$.

If r_{jk} denotes the entry in the j-th row of r_k, then

$$\begin{aligned}
\hat{u}_j^i &= \frac{p_j^i - v^\infty}{\|p_j^i - v^\infty\|} \\
&= \frac{\lambda^i(c_2 r_{j2} + c_3 r_{j3}) + \lambda_4^i c_4 r_{j4} + \cdots}{\|\lambda^i(c_2 r_{j2} + c_3 r_{j3}) + \lambda_4^i c_4 r_{j4} + \cdots\|} \\
&= \frac{(c_2 r_{j2} + c_3 r_{j3}) + \frac{\lambda_4^i}{\lambda^i} c_4 r_{j4} + \cdots}{\|(c_2 r_{j2} + c_3 r_{j3}) + \frac{\lambda_4^i}{\lambda^i} c_4 r_{j4} + \cdots\|}
\end{aligned}$$

In the limit as $i \to \infty$,

$$\hat{u}_j^\infty = \lim_{i\to\infty} \hat{u}_j^i = \frac{c_2 r_{j2} + c_3 r_{j3}}{\|c_2 r_{j2} + c_3 r_{j3}\|}. \tag{14}$$

Equation 14 implies that each of the limiting unit vectors $\hat{u}_j^\infty$, $j = 2, ..., m$ is a linear combination of the vectors c_2 and c_3. All the vectors $\hat{u}_j^\infty$ must therefore lie in the plane spanned by c_2 and c_3. The normal vector N^∞ we seek is therefore $c_2 \times c_3$. □

Again using a discrete Fourier transform technique, one can show that for Catmull-Clark surfaces,

$$\lambda := \lambda_2 = \lambda_3 = \frac{4 + A_n}{16}$$

$$c_2 = \sum_j A_n \cos(\frac{2\pi j}{n}) e_j^1 + (\cos(\frac{2\pi j}{n}) + \cos(\frac{2\pi(j+1)}{n})) f_j^1$$

where

$$A_n = 1 + \cos(\frac{2\pi}{n}) + \cos(\frac{\pi}{n})\sqrt{2(9 + \cos(\frac{2\pi}{n}))}.$$

The vector c_3 is obtained from c_2 by replacing e_j^1 with e_{j+1}^1 and f_j^1 with f_{j+1}^1.

B Integrating the fairness functional

In this appendix, we consider the problem of evaluating the fairness norm of Equation 7 for a patch whose local control mesh P contains an extraordinary point, such as the one shown in Figure 2(b). As motivated in Section 4, we will ultimately evaluate only the non-divergent part of the thin plate energy corresponding to the deviation of P from its regular component P'. As we show below, it is not necessary to compute P' explicity, so we will for the time being evaluate the energy of P.

The quadratic form $\mathbf{K}$ referred to in Equation 6 can be written as a weighted sum of two quadratic forms $\mathbf{K}_m$ and $\mathbf{K}_p$, representing the membrane and plate energies, respectively for a bicubic patch:

$$\mathbf{K} = \alpha\mathbf{K}_m + \beta\mathbf{K}_p.$$

Let $E(n, P, j)$ denote the fairness norm of Equation 7 integrated over a patch containing at most one extraordinary point of order n whose local mesh is described by the column vector of control points P, and whose level of subdivision is j. As outlined in Section 4, when $n \neq 4$, we evaluate $E(n, P, j)$ by splitting the patch into four subpatches, three of which are ordinary (shown in gray in Figure 2), and one of the same form as the original. the This leads to the following recurrence relation for $E(n, P, j)$:

$$\begin{aligned} E(4, P, j) &= P^T \left(\alpha \mathbf{K}_m + 4^j \beta \mathbf{K}_p\right) P \\ E(n, P, j) &= \sum_{k=1}^{3} E(4, \mathbf{\Omega}_k P, j+1) + E(n, \mathbf{\Omega}_4 P, j+1) \end{aligned}$$

where $\mathbf{\Omega}_1, \mathbf{\Omega}_2, \mathbf{\Omega}_3$ are matrices that carry P into the local meshes for the ordinary (shaded) subpatches, and where $\mathbf{\Omega}_4$ is the matrix that carries P into the local mesh for the remaining (unshaded) extraordinary subpatch.

The factor of 4^j in front of $\mathbf{K}_p$ reflects the change of integration variables when a patch is subdivided j times. The choice of powers of 4 is somewhat arbitrary. It corresponds to the parametrization assigned to the bicubic subpatches created when the extraordinary patch is subdivided. We have chosen powers of 4 since it is the correct factor for bicubic patches. We are, however, currently experimenting with methods to select this factor based on n.

The above recurrence can be unrolled to produce an infinite series for $E(n, P, 0)$:

$$E(n, P, 0) = \sum_{j=1}^{\infty} \sum_{k=1}^{3} E(4, \mathbf{\Omega}_k \mathbf{\Omega}_4^{j-1} P, j)$$

which can be written as

$$E(n, P, 0) = P^T \mathbf{K}_n P$$

where

$$\mathbf{K}_n := \sum_{j=1}^{\infty} (\mathbf{\Omega}_4^{j-1})^T (\bar{\mathbf{K}}_m + 4^j \bar{\mathbf{K}}_p) \mathbf{\Omega}_4^{j-1},$$

and where

$$\bar{\mathbf{K}}_m := \sum_{k=1}^{3} \alpha \mathbf{\Omega}_k^T \mathbf{K}_m \mathbf{\Omega}_k,$$

$$\bar{\mathbf{K}}_p := \sum_{k=1}^{3} \beta \mathbf{\Omega}_k^T \mathbf{K}_p \mathbf{\Omega}_k.$$

The limiting value of the series can be found by expanding $\mathbf{\Omega}_4$ in its basis of eigenvectors:

$$\mathbf{\Omega}_4 = \mathbf{X} \mathbf{\Lambda} \mathbf{X}^{-1}$$

where $\mathbf{\Lambda}$ is a diagonal matrix containing the eigenvalues of $\mathbf{\Omega}_4$, and where the columns of $\mathbf{X}$ are the corresponding right eigenvectors. Without loss of generality we can assume that the eigenvalues appear in decreasing order down the diagonal. $\mathbf{K}_n$ can now be written as

$$\begin{aligned} \mathbf{K}_n = \mathbf{X}^{-T} \underbrace{\left\{ \sum_{j=1}^{\infty} \mathbf{\Lambda}^{j-1} \mathbf{X}^T \bar{\mathbf{K}}_m \mathbf{X} \mathbf{\Lambda}^{j-1} \right\}}_{\tilde{\mathbf{K}}_m} \mathbf{X}^{-1} + \\ \mathbf{X}^{-T} \underbrace{\left\{ \sum_{j=1}^{\infty} 4^j \mathbf{\Lambda}^{j-1} \mathbf{X}^T \bar{\mathbf{K}}_p \mathbf{X} \mathbf{\Lambda}^{j-1} \right\}}_{\tilde{\mathbf{K}}_p} \mathbf{X}^{-1}. \end{aligned}$$

Since $\mathbf{\Lambda}$ is diagonal, the ab-th entry of $\tilde{\mathbf{K}}_m$ is

$$(\tilde{\mathbf{K}}_m)_{ab} = (\mathbf{X}^T \bar{\mathbf{K}}_m \mathbf{X})_{ab} \sum_{j=1}^{\infty} (\mathbf{\Lambda}_{aa})^{j-1} (\mathbf{\Lambda}_{bb})^{j-1}.$$

The above series is geometric, so if $\mathbf{\Lambda}_{aa} \mathbf{\Lambda}_{bb} < 1$, it converges to

$$(\tilde{\mathbf{K}}_m)_{ab} = \frac{(\mathbf{X}^T \bar{\mathbf{K}}_m \mathbf{X})_{ab}}{1 - \mathbf{\Lambda}_{aa} \mathbf{\Lambda}_{bb}}.$$

Using arguments as in appendix A, it can be shown that the largest eigenvalue of $\mathbf{\Omega}_4$ is one, meaning that the product $\mathbf{\Lambda}_{aa} \mathbf{\Lambda}_{bb}$ is at most one, and this occurs only when $a = b = 1$. The membrane energy is invariant under translation, which is reflected in the fact that $(\mathbf{X}^T \bar{\mathbf{K}}_m \mathbf{X})_{11}$ is zero; hence $(\tilde{\mathbf{K}}_m)_{11} = 0$.

A similar analysis for $\tilde{\mathbf{K}}_p$ shows that

$$(\tilde{\mathbf{K}}_p)_{ab} = 4(\mathbf{X}^T \bar{\mathbf{K}}_p \mathbf{X})_{ab} \sum_{j=1}^{\infty} 4^{j-1} (\mathbf{\Lambda}_{aa})^{j-1} (\mathbf{\Lambda}_{bb})^{j-1}.$$

Thus, $(\tilde{\mathbf{K}}_p)_{ab}$ is finite whenever $4\mathbf{\Lambda}_{aa} \mathbf{\Lambda}_{bb} < 1$. The factor $4\mathbf{\Lambda}_{aa} \mathbf{\Lambda}_{bb}$ can be shown to be one or larger when $1 \leq a, b \leq 3$. Just as for the membrane energy, the 11 entry poses no difficulty since $(\mathbf{X}^T \bar{\mathbf{K}}_p \mathbf{X})_{11} = 0$, indicating that the thin plate energy is invariant under translation.

The remaining 8 entries of $\tilde{\mathbf{K}}_p$ are unbounded for $n > 4$. When $n = 4$ (i.e., the ordinary case), $4\mathbf{\Lambda}_{aa} \mathbf{\Lambda}_{bb} = 1$, yet we know that the entries of $\tilde{\mathbf{K}}_p$ are finite since bicubic patches have finite thin plate energy. We therefore conclude that for $n = 4$, $(\tilde{\mathbf{K}}_p)_{ab} = 0$ for $1 \leq a, b \leq 3$. This reflects the fact that regular control meshes have zero thin plate energy. To generalize this idea to arbitrary n, we simply set the remaining 8 divergent terms to zero, which is equivalent to evaluating the norm on $P - P'$.

To summarize, the quadratic form related to the thin plate energy is taken to be

$$(\tilde{\mathbf{K}}_p)_{ab} = \begin{cases} \frac{4(\mathbf{X}^T \bar{\mathbf{K}}_p \mathbf{X})_{ab}}{1 - 4\mathbf{\Lambda}_{aa} \mathbf{\Lambda}_{bb}} & \text{if } 4\mathbf{\Lambda}_{aa} \mathbf{\Lambda}_{bb} < 1 \\ 0 & \text{otherwise} \end{cases}$$

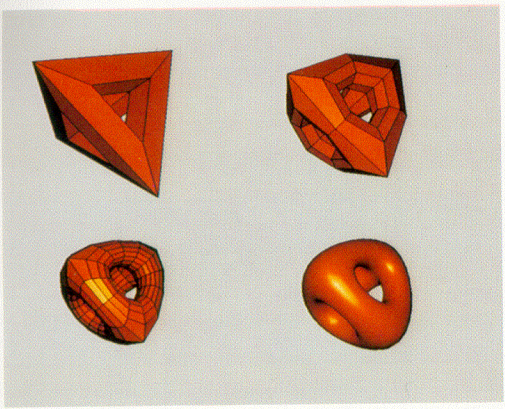

Figure 3: Upper Left: Tetrahedral mesh with holes. Upper Right: The mesh after one Catmull-Clark subdivision. Lower Left: The mesh after two subdivisions. Lower Right: The limit surface.

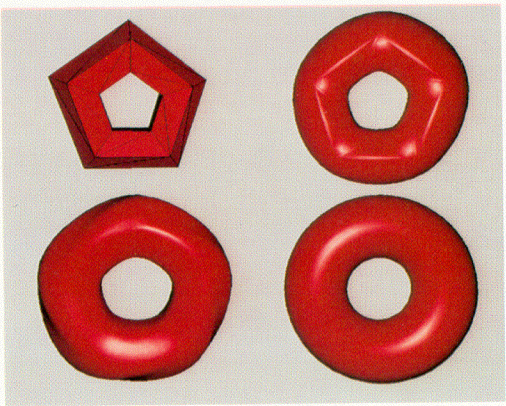

Figure 4: Interpolating a coarsely polygonized torus. Upper left: original mesh. Upper right: Shirman-Séquin interpolation[14]. Lower left: Interpolating Catmull-Clark surface. Lower right: Faired interpolating Catmull-Clark surface.

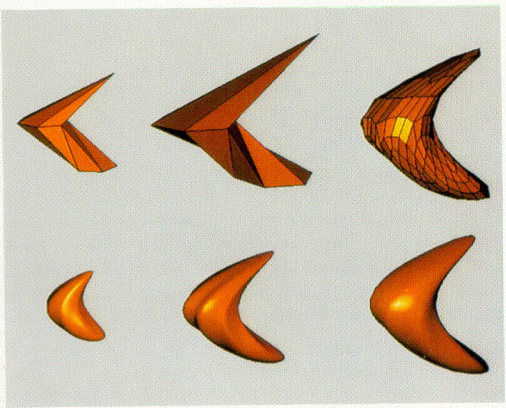

Figure 5: Top row: Original mesh, Interpolating mesh, Faired interpolating mesh. Bottom row: Corresponding Catmull-Clark surfaces. Interpolation introduces wiggles which are removed by fairing.

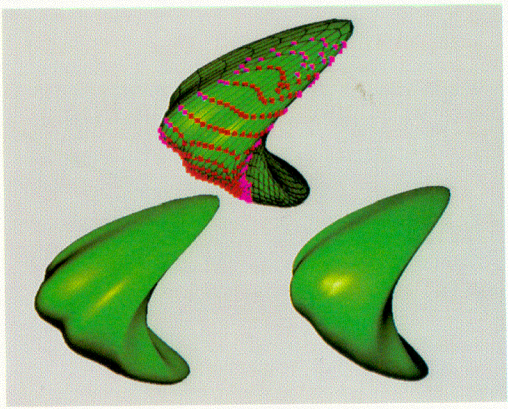

Figure 6: Lower left: unfaired interpolating surface. Upper center: Interactive fairing. Red vertices are allowed to move. Magenta vertices influence the minimization, but remain fixed. Lower right: Result after fairing.

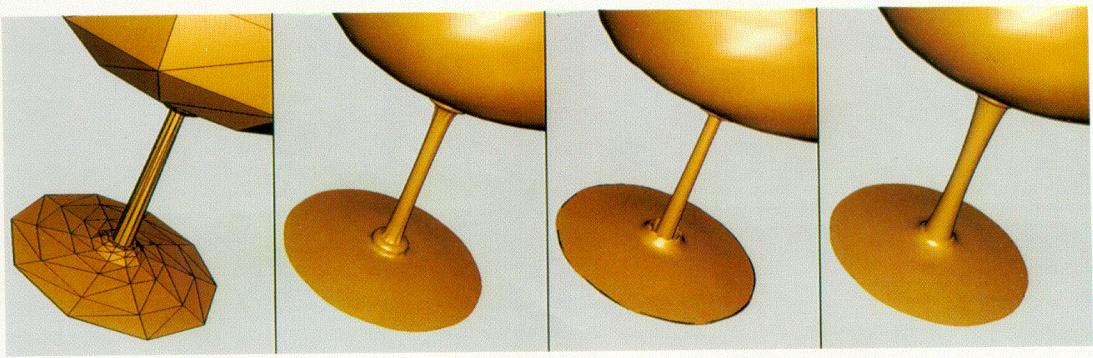

Figure 7: From left to right: Original goblet mesh containing 190 vertices. Ordinary Catmull-Clark surface (approximating). Interpolating Catmull-Clark surface. Faired interpolating Catmull-Clark surface. The far right surface interpolates the original mesh without the artifacts present in the middle two surfaces.