1990

Surface Fitting Using Implicit Algebraic Surface Patches

Chandrajit Bajaj

Report Number:
90-1001

SURFACE FITTING USING IMPLICIT ALGEBRAIC SURFACE PATCHES

Chandrajit Bajaj

CSD-TR-1001
August 1990

1 Introduction

Interpolation and least-squares approximation provide efficient ways of generating C^k-continuous meshes of surface patches, necessary for the construction of accurate computer geometric models of solid physical objects [see for e.g. [8, 7]. Two surfaces $f(x, y, z) = 0$ and $g(x, y, z) = 0$ meet with C^k-continuity along a curve C if and only if there exists functions $\alpha(x, y, z)$ and $\beta(x, y, z)$ such that all derivatives upto order k of $\alpha f - \beta g$ equals zero [see for e.g.[62]]. C^k-continuity of two surface patches follows if the above condition is true along the common boundary curves between the two patches.

This paper surveys the use of low degree, implicitly defined, algebraic surfaces and surface patches in three dimensional real space $\mathbb{R}^3$ for various scattered data C^k-fitting problems. The use of low degree algebraic surface patches to construct models of physical objects stems from the advantage of faster computations in subsequent geometric model manipulation operations such as computer graphics display, animation, and physical object simulations, see for e.g. [10].

Why algebraic surfaces ? A real algebraic surface S in $\mathbb{R}^3$ is implicitly defined by a single polynomial equation $\mathcal{F} : f(x, y, z) = 0$, where coefficients of f are over the real numbers $\mathbb{R}$. Manipulating polynomials, as opposed to arbitrary analytic functions, is computationally more efficient. Furthermore algebraic surfaces provide enough generality to accurately model almost all complicated rigid objects.

Why implicit representations ? While all real algebraic surfaces have an implicit definition $\mathcal{F}$ only a small subset of these real surfaces can also be defined parametrically by the triple $\mathcal{G}(s, t)$: $(x = G_1(s, t), y = G_2(s, t), z = G_3(s, t))$ where each G_i, $i = 1, 2, 3$, is a rational function (ratio of polynomials) in s and t over $\mathbb{R}$. The primary advantage of the implicit definition $\mathcal{F}$ is the closure properties of the complete class of algebraic surfaces under modeling operations such as intersection, convolution, offset, blending, etc. The smaller class of parametrically defined algebraic surfaces $\mathcal{G}(s, t)$ are not closed under any of the operations listed before. Closure under modeling operations allow cascading repetitions[1] without any need of approximation. Furthermore, designing with a larger class of surfaces leads to better possibilities (as we show here) of being able to satisfy the same geometric design constraints with much lower degree algebraic surfaces. The implicit representation of smooth algebraic surfaces also naturally yields half-spaces $\mathcal{F}^+ : f(x, y, z) \geq 0$ and $\mathcal{F}^- : f(x, y, z) \leq 0$, a fact quite useful for intersection and offset modeling operations. Finally, most prior approaches to interpolation and least-squares scattered data fitting, have focused on the parametric representation of surfaces [29, 50, 57, 64]. Our aim here is to exhibit that implicitly defined algebraic surfaces are also very appropriate for geometric surface design.

Additional Notation and Definitions : A real algebraic space curve can be implicitly defined as the common intersection of two or more real algebraic surfaces $C : (f_1(x, y, z) = 0, f_2(x, y, z) = 0, f_3(x, y, z) = 0, \ldots)$. A smaller class of *rational* algebraic space curves can also be represented by the triple $\mathcal{H}(s) : (x = H_1(s), y = H_2(s), z = H_3(s))$, where H_1, H_2 and H_3 are rational functions in s over $\mathbb{R}$. Whenever we consider the special case of a rational space curve, we assume that the

[1] The output of one operation acts as the input to another operation

curve is smooth and only singly defined under the parameterization map, i.e., each triple of values for (x, y, z), corresponds to a single value of s.

The "normal" N_p of a point p is an arbitrary nonzero vector associated with p. N_p defines a unique plane containing p. The "normal" N_C of a curve C is a 1-dimensional set of vectors, one vector associated with each point p on C, and orthogonal to the tangent vector at p. We assume are curves are smooth i.e. nonsingular, though this is not a necessary requirement. Finally, a surface patch is defined as a smooth, connected 2-dimensional region of a surface bounded by a single cycle of curve segments.

Problem Descriptions :

1. *C^k Interpolation Surface Fit*: Construct a single real algebraic surface S which C^k interpolates a collection of l points $\mathbf{p}_i$ in $\mathbb{R}^3$ with associated fixed "normal" unit vectors $\mathbf{m}_i$, and m given space curves C_j in $\mathbb{R}^3$, possibly with associated "normal" unit vectors $\mathbf{n}_j$ and additionally upto $(k-1)^{th}$ order derivatives[2] of $\mathbf{n}_j$, varying along the entire span of the curves. Assume that any of the vectors $\mathbf{m}_i$ and $\mathbf{n}_j$ or their derivatives are never identically zero, a phenomenon that occurs at singularities. By C^k-interpolation we shall mean that the interpolating surface S contains each of the points and curves and furthermore has its gradient together with its $1 \ldots (k-1)^{th}$ order derivatives, respectively, in the same direction as the specified "normal" vectors and its derivatives along the entire span of the C_j's. This is one natural generalization into space of the usual two dimensional Hermite interpolation, applied to fitting curves through point data and equating derivatives at those points.

2. *C^k Least-Squares Approximate Surface Fit*: Construct a real algebraic surface S, which C^{k-1} interpolates a collection of points $\mathbf{p}_i$ in $\mathbb{R}^3$ and given space curves C_j in $\mathbb{R}^3$ as before, with associated unit "normal" vectors and its $1 \ldots (k-2)^{th}$ order derivatives, and additionally minimizes the Euclidean 2-norm of the difference of the $(k-1)^{th}$ order derivative of S's gradient and the $(k-1)^{th}$ order derivative of the specified unit "normal" vectors, on the same collection of points and space curves. This is a natural generalization of ordinary C^0 least-squares approximation (the case $k = 0$) which minimizes only the sum of the squares of the distances of the solution from a collection of points or curves.

3. *C^k Interpolation and C^l Least Squares Fit with Surface Patches* : Construct a mesh of real algebraic surface patches S_i, which C^k interpolates a collection of points $\mathbf{p}_i$ in $\mathbb{R}^3$ and given space curves C_j in $\mathbb{R}^3$, with associated "normal" unit vectors and their derivatives, varying along the entire span of the curves and C^l least squares approximates a collection of points $\mathbf{q}_i$ in $\mathbb{R}^3$ and given space curves D_j in $\mathbb{R}^3$ with associated "normal" unit vectors and their derivatives, varying along the entire span of the curves. The set of points $\mathbf{p}_i$ and $\mathbf{q}_i$ are not necessarily disjoint and neither are the set of curves C_j and D_j.

[2]The emphasis being algebraic space curves, the "normals" and higher order derivatives along curves are restricted to polynomials of some degree.

4. *Triangulated Data Fit with Surface Patches :*

   Given a collection of Z-values and derivatives over a triangulation $\mathcal{T}$, construct a mesh of real algebraic surface patches S_i, which C^k interpolates the collection of points $\mathbf{p}_i = (x_i, y_i, z_i)$ in $\mathbb{R}^3$ and C^l least squares approximates the collection of points $\mathbf{q}_j = (x_j, y_j, z_j)$ in $\mathbb{R}^3$. The set of points $\mathbf{p}_i$ and $\mathbf{q}_i$ are not necessarily disjoint and $\mathcal{T}$ may be either an $X - Y$ (2D) or an $X - Y - Z$ (3D) triangulation of the entire collection of points.

5. *Interactive Shape Control of Implicit Surface Families :*

Interactively control the shape of an interpolating or approximating implicit surface by selecting appropriate instances from a p-parameter family of solution surfaces.

Paper Outline:

The rest of the paper is structured as follows. Each of the subsequent sections 2 – 6 is devoted to one of the above problems, and summarizes various recent approaches to implicit surface fitting for the appropriate problem. The section then details a recent result, which the author is most familiar with, and provides examples to clarify the algorithm presented.

2 C^k Interpolation Surface Fit

Problem

Construct a single real algebraic surface S which C^k interpolates a collection of l points $\mathbf{p}_i$ in $\mathbb{R}^3$ with associated fixed "normal" unit vectors $\mathbf{m}_i$, and m given space curves C_j in $\mathbb{R}^3$, possibly with associated "normal" unit vectors $\mathbf{n}_j$ and additionally upto $(k-1)^{th}$ order derivatives

Summary of Approaches

There has been extensive prior work in interpolatory or exact surface fitting through scattered data. Much of it has either concentrated on polynomial parametric (and ocassionally rational parametric) surface fitting through scattered point data in 3D. see for e.g., the surveys by Alfeld [2], Bohm et. al. [20], Franke [32], Sabin [56]. Exact fitting of curves (primarily conics) has been considered by several authors, see for eg [15, 19, 35, 49, 58]. An exposition of exact C^0 fitting of implicitly defined algebraic surfaces through given data points, is presented in [54]. Characterizations of C^0 surface fits of points and curves using implicitly defined algebraic surfaces is also given by [60]. Other approaches to parametric surface fitting and transfinite interpolation are also mentioned in that paper, as well as in [50, 64]. Paper [12, 14] generalizes the results of [54, 60]. It provides conditions for exact C^k fits of implicitly defined algebraic surfaces through given points and space curves together with derivative information ("normals") along the curves.

Recent Results

Bajaj and Ihm in [11], present a simple constructive characterization of the real algebraic surface which C^1 interpolates any given number of points and algebraic space curves, with associated "normal" directions. This characterization, called *Hermite interpolation*, deals with the containment and matching normals at the points or varying along the entire span of the space curves.

The input for Hermite interpolation is a description of the properties of a surface to be designed in terms of a combination of points and curves, possibly associated with "normal" directions. For an algebraic surface S of degree n, C^1-interpolation generates a homogeneous linear system $\mathbf{M_I x} = \mathbf{0}$ where $\mathbf{x}$ is a $\binom{n+3}{3}$-vector[3] of the coefficients of the algebraic surface S. All nontrivial vectors, if any, in the nullspace of $\mathbf{M_I}$ forms a family of all the surfaces, satisfying the given description. The coefficients of the family of surfaces are expressed in terms of p-parameters where p is the rank of the nullspace.

In C^1-interpolation, smoothness is achieved by making the normals of tangent planes of the surface to be designed identical to those of given points or curves. For some applications of modeling, such as design of the body of an airplane, however, more than tangent plane smoothness is desirable. This concept of smoothness is generalized by defining a higher order of geometric continuity. DeRose [29] gives such a definition between parametric surfaces, where two surfaces F_1 and F_2 meet with order k geometric continuity (concisely stated as C^k continuity) along a curve C if and only if there exist local reparameterizations F_1' and F_2' of F_1 and F_2, respectively, such that all partial derivatives of F_1' and F_2' up to degree k agree along C. Warren [62] formulates an intuitive definition of C^k continuity between implicit surfaces as follows :

Definition 2.1 *Two algebraic surfaces $f(x,y,z) = 0$ and $g(x,y,z) = 0$ meet with C^k continuity at a point p or along an irreducible algebraic curve C if and only if there exists two polynomials $a(x,y,z)$ and $b(x,y,z)$, not identically zero at p or along C, such that all derivatives of $a \cdot f - b \cdot g$ up to degree k vanish at p or along C.*

This formulation is *more general* than just making all the partials of $f(x,y,z) = 0$ and $g(x,y,z) = 0$ agree at a point or along a curve. For example, consider the intersection of the cone $f(x,y,z) = xy - (x+y-z)^2 = 0$ and the plane $g(x,y,z) = x = 0$ along the line defined by two planes $x = 0$ and $y = z$. It is not hard to see that these two surfaces meet smoothly along the line since the normals to $f(x,y,z) = 0$ at each point on the line are scalar multiples of those to $g(x,y,z) = 0$. But, this scale factor is a function of z. Situations like these are corrected by allowing multiplication by certain polynomials, not identically zero along a intersection curve. Note that multiplication of a surface by polynomials nonzero along a curve does not change the geometry of the surface in the neighborhood of the curve. Garrity and Warren in [34] also prove that this notion of rescaling C^k-continuity is equivalent to other k^{th} order derivative continuity measures as well as to reparameterization continuity for parametric surfaces. In [12], Bajaj and Ihm show how to form a C^1-interpolation matrix $\mathbf{M_I}$ and proved that using this one is able to construct all surfaces meeting each other with rescaling C^1-continuity. However, even though one is currently unable to translate geometric specifications for C^k-continuity ($k \geq 2$) into a matrix $\mathbf{M_I}$ whose nullspace captures *all* C^k continuous surfaces, from the theorem below one can generate an interpolation matrix $\mathbf{M_I}$ whose nullspace captures an interesting proper subset of the whole class.

Theorem 2.1 ([14]) *If surfaces $g(x,y,z) = 0$ and $h(x,y,z) = 0$ intersect transversally in along an irreducible curve C, then any algebraic surface $f(x,y,z) = 0$ that meets $g(x,y,z) = 0$ with C^k-continuity along C must be of the form $f(x,y,z) = \alpha(x,y,z)g(x,y,z) + \beta(x,y,z)h^{k+1}(x,y,z)$.*

[3] There are $\binom{n+3}{3}$ coefficients in $f(x,y,z)$ of degree n

If $g(x,y,z) = 0$ and $h(x,y,z) = 0$ share no common components at infinity, then the degree of $\alpha(x,y,z)g(x,y,z) \le$ degree of $f(x,y,z)$ and the degree of $\beta(x,y,z)h^{k+1}(x,y,z) \le$ degree of $f(x,y,z)$.

For given curves C_i, $i = 1 \ldots l$ which are respectively the transversal intersection of given surfaces $g_i(x,y,z) = 0$ and $h_i(x,y,z) = 0$, a surface $f(x,y,z) = 0$ containing space curves C_i with C^k-continuity then can be constructively obtained by the relations

$$f(x,y,z) = \alpha_i(x,y,z)g_i(x,y,z) + \beta_i(x,y,z)h_i^{k+1}(x,y,z) \quad i = 1 \ldots l \tag{1}$$

Since the g_i and h_i are known surfaces, the unknown coefficients are those of f, α and β. Note, from the above theorem, for a possible interpolating surface f of degree n, both polynomials α and β are of bounded degree. From the relations in (1) one sees that these unknown coefficients form a system of linear equations, yielding the interpolation matrix $\mathbf{M_I}$. For the special case of parametric space curves with parametric "normal" and derivative information, the above technique can also be adapted to provide C^k-continuous algebraic surface fits. Here using C^1 interpolation [12], implicit surfaces are first constructed which contain the parametric curve as well as have matching "normals" and derivative information. These implicit surfaces are then used above, to generate matrix $\mathbf{M_I}$ for C^k-continuous fits.

Examples

Ex 2.1 *A Quartic Surface for a C^1 Blend of the Corner of a Table*

The edges of the table corner are given by : $C_1 : (y^2+z^2-25 = 0, x = 0)$, and $C_2 : (x^2+z^2-25 = 0, y = 0)$. Each curve is associated with a "normal" direction which is chosen in the same direction as the gradients of the side of table, the cylinder in C_1 and C_2. That is, $\mathbf{n}_1(x,y,z) = (0, 2y, 2z)$, and $\mathbf{n}_2(x,y,z) = (2x, 0, 2z)$.

The interpolation matrix $\mathbf{M_I}$ is of size 32×35 (32 linear equations and 35 coefficients for a quartic surface) whose rank is 24. The nullspace of $\mathbf{M_I}$ is of dimension 11 represented by a family of quartic surfaces which blend the corner $f(x,y,z) = r_1 z^4 + (r_2 y + r_6 x + 5r_4)z^3 + (r_3 y^2 + (r_7 x + 5r_8)y + r_{10}x^2 + 5r_{11}x - 25r_9 - 25r_1)z^2 + (r_2 y^3 + (r_6 x + 5r_4)y^2 + (r_2 x^2 - 25r_2)y + r_6 x^3 + 5r_4 x^2 - 25r_6 x - 125r_4)z + (r_3 - r_1)y^4 + (r_7 x + 5r_8)y^3 + (r_5 x^2 + 5r_{11}x - 25r_9 - 25r_3 + 25r_1)y^2 + (r_7 x^3 + 5r_8 x^2 - 25r_7 x - 125r_8)y + (r_{10} - r_1)x^4 + 5r_{11}x^3 + (-25r_9 - 25r_{10} + 25r_1)x^2 - 125r_{11}x + 625r_9$.

An instance $f(x,y,z) = -1250 - x^4 - y^4 - x^2z^2 - y^2z^2 + 50z^2 + 75y^2 + 75x^2$ of this family is shown with the table in Figure 1. □

Ex 2.2 *A Quartic Interpolating Surface for a C^1 Join of Four Parallel Cylindrical Surfaces*

In this example, the lowest degree surface is constructed, which smoothly joins four truncated parallel circular cylinders defined by $CYL_1 : y^2 + z^2 - 1 = 0$ for $x \ge 2$, $CYL_2 : y^2 + z^2 - 1 = 0$ for $x \le -2$, $CYL_3 : (y-4)^2 + z^2 - 1 = 0$ for $x \ge 2$, and $CYL_4 : (y-4)^2 + z^2 - 1 = 0$ for $x \le -2$.

The C^1 interpolation technique shows that the minimum degree for such joining surface is 4, and finds a 2-parameter (one independent parameter) family of algebraic surfaces which is $f(x,y,z) = \frac{r_1}{14}z^4 + \frac{r_1}{7}y^2z^2 - \frac{4r_1}{7}yz^2 + r_1 z^2 + \frac{r_1}{14}y^4 - \frac{4r_1}{7}y^3 + r_1 y^2 + \frac{4}{7}r_1 y + \frac{14r_2+15r_1}{224}x^4 - \frac{14r_2+15r_1}{28}x^2 + r_2$.

An instance of this family ($r_1 = 392$, $r_2 = -868$) is shown in Figure 2.

Ex 2.3 *Cubic and Quartic Surfaces interpolating with C^2 and C^3 continuity*

Consider a space curve C defined by the two equations $f_1(x,y,z) = x^2 + 2y^2 + 2z^2 - 2 = 0$ and $f_2(x,y,z) = x = 0$. A cubic surface C_1 is constructed which interpolates C with G^2 continuity as follows: The general implicit equation of a cubic algebraic surface is given by $f_3(x,y,z) = ax^3+by^3+cz^3+dx^2y+exy^2+fx^2z+gxz^2+hy^2z+iyz^2+jxyz+kx^2+ly^2+mz^2+nxy+oyz+pxz+qx+ry+sz+t = 0$. Using relation (1) for G^2 continuity as given in Section 2, one obtains $f_3(x,y,z) = (r_1x+r_2y+r_3z+r_4)f_1(x,y,z)+r_5f_2(x,y,z)^3$ yielding the system of linear equations $a-r_1-r5 = 0$, $b-2r_2 = 0$, $c-2r_3 = 0$, $d-r2 = 0$, $e-2r_1 = 0$, $f-r3 = 0$, $g-2r_1 = 0$, $h-2r_3 = 0$, $i-2r_2 = 0$, $j = 0$, $k-r_4 = 0$, $l-2r_4 = 0$, $m-2r_4 = 0$, $n = o = p = 0$, $q+2r_1 = 0$, $r+2r_2 = 0$, $s+2r_3 = 0$, $t+2r_4 = 0$ in unknowns $a,\ldots,t$ and $r_1,\cdots,r_5$. For $r_1 = 1$, $r_2 = -1$, $r_3 = 1$, $r_4 = 1$, $r_5 = 2$, the cubic surface $f_3(x,y,z) = 2z^3-2yz^2+2xz^2+2z^2+2y^2z+x^2z-2z-2y^3+2xy^2+2y^2-x^2y+2y+3x^3+x^2-2x-2$ is shown in Figure 3.

In the same way, a quartic surface $f_4(x,y,z) = 16z^4-16yz^3+32xz^3+32z^3+16y^2z^2-16xyz^2-16yz^2+24x^2z^2+32xz^2-16y^3z+32xy^2z+32y^2z-8x^2yz+16yz+32x^3z+16x^2z-32xz-32z-9y^4-16xy^3-16y^3+16x^2y^2+32xy^2+16y^2-8x^3y-8x^2y+16xy+16y+24x^4+32x^3-8x^2-32x-16$ is constructed which meets f_3 with G^3 continuity along the curve defined by f_3 and $f_5(x,y,z) = y = 0$ as shown in Figure 4

□

Open Problems

1. Reduce implicit surface interpolation for higher geometric continuity to a linear system which captures all possible solutions ?
2. Investigate the relationship of the degrees and relative topology of the input curves with the rank of the interpolation matrices ?

3 C^k Least-Squares Approximate Surface Fit

Problem

Construct a real algebraic surface S, which C^{k-1} interpolates a collection of points $\mathbf{p}_i$ in $\mathbb{R}^3$ and given space curves C_j in $\mathbb{R}^3$ as before, with associated unit "normal" vectors and its $1\ldots(k-2)^{th}$ order derivatives, and additionally minimizes the Euclidean 2-norm of the difference of the $(k-1)^{th}$ order derivative of S's gradient and the $(k-1)^{th}$ order derivative of the specified unit "normal" vectors, on the same collection of points and space curves.

Summary of Approaches The concept of C^k least squares fitting and also through a mixture of point and curve 3D data is suprisingly novel and search of the past literature failed to reveal a suitable reference. Pratt [54] and some others [43] consider the traditional C^0 least-squares approximation problem using implicit algebraic surface for only scattered point data. In Bajaj, Ihm and Warren [14], a C^k interpolating/least-squares approximating implicit algebraic surface is found by solving a quadratic optimization problem constructed from given sets of 3D points and curves data. In this method, higher order derivative information of points and space curves as well as positional data, is interpolated and approximated to. For example, when a surface of some fixed degree doesn't have sufficient flexibility to C^1 interpolate a set of curves with normal directions, the tangential constraints are least-squares approximated after the positional constraints are exactly C^0 interpolated.

Recent Results

In C^k interpolation of the previous subsection, one seeks a nontrivial solution $\mathbf{x}$ which is the coefficient vector of an algebraic surface $f(x, y, z) = 0$. To use least-squares approximations for geometric design, one needs to define distance metrics which are meaningful and also computationally viable. As $f(x, y, z) = cf(x, y, z)$ for $c \neq 0$, the coefficients of f are first normalized so that $f(x, y, z)$ represents the equivalence class $\{cf(x, y, z) | c \neq 0\}$. There are infinite number of ways to normalize $f(x, y, z) = 0$. Bajaj, Ihm and Warren [14] choose to adopt *quadratic normalization*. This normalization has been extensively used in the approximate fitting of conic sections as well [1, 15, 19, 35, 49, 58], and has yielded computationally efficient algorithms. Quadratic normalization is of the form $\mathbf{x}^T \mathbf{M_N} \mathbf{x} = 1$ where $\mathbf{M_N}$ is a real symmetric matrix. In most cases the identity matrix $\mathbf{I}$ or a diagonal matrix $\mathbf{D}$ is used for the matrix $\mathbf{M_N}$.

Once normalization of the coefficients is done, one may use —$f(p)$— as a distance metric. This metric, called *the algebraic distance*, is straightforward to compute and in some cases, closely approximates the real geometric distance (the Euclidean distance between a point and a surface). Sampson [58] proposes the use $\frac{f(p)}{\nabla f(p)}$ as a distance measure (a nonalgebraic distance). Perhaps a better approximation is achievable in some cases, however, only at the enormous cost of iterative applications of least-squares approximation.

Least Squares Approximation can be directly used to control the geometric shape a solution interpolating surface. When the rank r of $\mathbf{M_I}$ of section 3 is less than $\binom{n+3}{3}$, the number of the unknown surface coefficients, there exists a family $f(x, y, z)$ of algebraic surfaces which satisfy the given geometric constraints and whose coefficients are expressed in terms of $p = \binom{n+3}{3} - r$ parameters. The problem of interactively selecting an instance from the solution family is addressed in section 6. Selecting an instance from the family is equivalent to assigning values to each of the p parameters. When there are p parameters to be instantiated, one may additionally specify a set of points, curves or even surfaces around the earlier given input data, which approximately describes the final surface to be designed. The final solution instance is computed via interpolation of the given input data and with least-squares approximation of the additional data set. In all these cases, from the matrix $\mathbf{M_I}$ one is easily able to construct a matrix $\mathbf{M_A}$ under the appropriate normalization $\mathbf{M_N}$, such that $\|\mathbf{M_A}\mathbf{x}\|^2 = \mathbf{x}^T {\mathbf{M_A}}^T \mathbf{M_A} \mathbf{x}$ is minimized. The normalization eliminates certain columns of the matrix $\mathbf{M_I}$ yielding an overdetermined reduced matrix $\mathbf{M_A}$ in the standard way [43].

Bajaj, Ihm and Warren [14] provide an efficient algorithm based on the orthogonal decomposition of the matrix and computation of eigenvalues and eigenvectors. As a means of computing the nullspace of the system, one uses the *QR method* based upon the Householder's transformation [36, 43]. In order to correctly decide the rank of the matrix during the Householder's transformation, the elements of a lower right part of a matrix are checked for zeros at each step.

Examples

Ex 3.1 *A Quartic Surface with C^1 Interpolation and C^0 Least Squares Approximation*

In this example, a quartic surface $f(x,y,z) = 0$ C^1 interpolates curves on the four cylinders described by $CYL_1 : y^2 + z^2 - 1 = 0$ for $x \geq 2$, $CYL_2 : y^2 + z^2 - 1 = 0$ for $x \leq -2$, $CYL_3 : x^2 + y^2 - 1 = 0$ for $z \geq 2$, and $CYL_4 : x^2 + y^2 - 1 = 0$ for $z \leq -2$. As a by-product during interpolation, it is found that degree 4 is the minimum required. For this interpolation, $\mathbf{M_I}$ is of size 64×35 (64 linear equations and 35 coefficients with rank 33, yielding a 2-parameter family of quartic surfaces satisfying the C^1 interpolation constraints. A specific member of the 2-parameter family of surfaces is next selected, via C^0 least-squares approximation from a collection of new data points. For the normalization $\mathbf{M_N}$, an identity matrix $\mathbf{I_{35}}$ is used. To illustrate the shaping effect of the approximation, two independent sets of data points are used: $S_1 = \{(0,1.75,0),(0,-1.75,0),(-1,1.25,0),(-1,-1.25,0),(1,1.25,0),(1,-1.25,0)\}$ and $S_2 = \{(0,1.25,0),(0,-1.25,0),(-0.5,1.125,0),(-0.5,-1.125,0),(0.5,1.125,0),(0.5,-1.125,0)\}$ (See Figure 5(a)(b)).

For the least-squares approximation with normalization, the eigenvalues and eigenvectors for S_1 and S_2 are computed. As a result, one obtains $\lambda_{mins_{S_1}} = 1.2546390$, $\lambda_{mins_{S_2}} = 0.6439209$, $\mathbf{y}_{mins_{S_1}} = [-0.1111540, 0.9938032]^t$, and $\mathbf{y}_{mins_{S_2}} = [0.01853292, 0.9998283]^t$. The corresponding surfaces after normalization are shown in Figure 6(a)(b). □

Ex 3.2 *A Quartic Surface with C^0 Interpolation and C^1 Least Squares Approximation*

Figure 7 shows two quartic triangluar patches which meet each other with C^0 interpolation continuity and are made as C^1 as possible along the entire common curve via least-squares approximation. The containment and tangency constraints for each patch are generated for three boundary conic curves with associated quadratic normals. It can be shown that degree 5 is the lowest possible degree of a surface which would C^1 interpolate the 3 boundary curves. □

Open Problems

1. Produce well-conditioned matrices $\mathbf{M_I}$ and ma for interpolation and approximation by appropriate choice of basis for the implicit surfaces.

2. Investigate the relationship between stability of computation and topology of the geometric data ?

4 C^k Interpolation and C^l Least Squares Fit with Surface Patches

Problem

Construct a mesh of real algebraic surface patches S_i, which C^k interpolates a collection of points $\mathbf{p}_i$ in $\mathbb{R}^3$ and given space curves C_j in $\mathbb{R}^3$, with associated "normal" unit vectors and their derivatives, varying along the entire span of the curves and C^l least squares approximates a collection of points $\mathbf{q}_i$ in $\mathbb{R}^3$ and given space curves D_j in $\mathbb{R}^3$ with associated "normal" unit vectors and their derivatives, varying along the entire span of the curves. The set of points $\mathbf{p}_i$ and $\mathbf{q}_i$ are not necessarily disjoint and neither are the set of curves C_j and D_j.

Summary of Approaches

Solving a linear system of equations plays a key role in C^k interpolation and approximation of the previous two sections. This section presents another approach of algebraic surface design [9], where a *nonlinear* system of polynomial equations needs to be solved. The emphasis here is on constructing C^k continuous meshes of implicit surface patches. Such "smooth" meshing has been largely addressed by [51, 39, 57] amongst others, using the Bézier representations of functional or parametric surfaces.

Recent Results

The technique of [9] is primarily based on Bezout's surface intersection theorem see [65]

Theorem 4.1 *If an algebraic surface S of degree n intersects an algebraic surface T of degree m in a curve of degree d with intersection multiplicity i, then $i * d \leq nm$.*

and a theorem from [14]

Theorem 4.2 *If surfaces f(x,y,z) = 0 and g(x,y,z) = 0 intersect transversally in a single irreducible curve[4] C, then any algebraic surface h(x,y,z) = 0 contains C with C^k continuity must be of the form $h(x,y,z) = \alpha(x,y,z)f(x,y,z) + \beta(x,y,z)g^{k+1}(x,y,z)$. Furthermore, the degree of $\alpha(x,y,z)f(x,y,z) \leq$ degree of h(x,y,z) and the degree of $\beta(x,y,z)g^{k+1}(x,y,z) \leq$ degree of h(x,y,z).*

Another theorem that is required, relates continuity with the intersection multiplicity of smooth algebraic surfaces, see [33, 34].

Theorem 4.3 *Two smooth algebraic surfaces $S_1 : f(x,y,z) = 0$ and $S_2 : g(x,y,z) = 0$ meet with C^k continuity along a curve C if and only if S_1 and S_2 intersect with multiplicity $k+1$ along C.*

From theorem 4.2 one obtains the following special case lemma

Lemma 4.1 *Let $S : f(x,y,z) = 0$ be an irreducible quadric surface, and $Q : q(x,y,z) = 0$ be a plane which intersects S in a conic C. Then, another quadric surface $S_1 : f_1(x,y,z)$ is tangent to S along C if and only if there exists nonzero constants α, β (possibly complex) such that $f_1 = \alpha f + \beta q^2$.*

Since one is interested in surface fitting with real surfaces, α and β are restricted to be real numbers. A related theorem can be derived for the quadric surface interpolation of two conics in space.

Lemma 4.2 *Consider quadrics $S_1 : f_1 = 0$. $S_2 : f_2 = 0$ and planes $Q_1 : q_1 = 0$, $Q_2 : q_2 = 0$. Let $C_1 : (f_1 = 0, q_1 = 0)$ and $C_2 : (f_2 = 0, q_2 = 0)$ be two conics in space. Then C_1 and C_2 can be Hermite interpolated by a quadric surface S if and only if there exist nonzero constants α_1, α_2, β_1, and β_2 (possibly complex) such that $\alpha_1 f_1 + \beta_1 q_1^2 - \alpha_2 f_2 - \beta_2 q_2^2 = 0$.*

[4] More precisely surfaces f(x,y,z) =0 and g(x,y,z) =0 intersect properly and share no common components at infinity

Proof: Trivial. (Just apply Lemma 4.1 twice.) ♠

This lemma is constructive, in that, it again yields a system of linear equations and a direct way of computing a C^1-interpolating quadric surface. Furthermore a solution to the above equations, linear in the α's and β's, exists if and only if such an interpolating quadric surface exists. Again, when real surfaces are favorable, one requires α_1, α_2, β_1, and β_2 to be real numbers.

Ex 4.1 *Suppose* $C_1 : (x^2 + z^2 - 1 = 0, 3x + y = 0)$, *and* $C_2 : (y^2 + z^2 - 1 = 0, x + 3y = 0)$. *The following equation is obtained from Lemma 4.2:* $(\alpha_1 + 9\beta_1 - \beta_2)x^2 + (\beta_1 - \alpha_2 - 9\beta_2)y^2 + (\alpha_1 - \alpha_2)z^2 + (6\beta_1 - 6\beta_2)xy + (\alpha_1 - \alpha_2) = 0$. *This implies* $\alpha_1 = \alpha_2$, $\beta_1 = \beta_2$, $\alpha_1 = -8\beta_1$. *When* $\alpha_1 = -8$ *and* $\beta_1 = 1$, *the interpolating surface is* $x^2 + y^2 - 8z^2 + 6xy + 8 = 0$.

In the Lemma 4.2 and the example, the two conics on the given quadric surfaces, S_1 and S_2, were fixed. If one has freedom to choose different intersecting planes Q_1 and Q_2 then one is able to find a family of quadric interpolating surfaces. In this case, the equations of planes Q_1 and Q_2 would have unknown coefficients and the use of Lemma 4.2 would result in a nonlinear system of equations, linear in terms of α_1, α_2, β_1 and β_2, and quadratic in terms of the unknowns of the plane's equations.

Now, rather than trying to find a single quadric surface, one can also extend the above Lemma 4.2, to construct two or more quadrics which smoothly contain two given conics in space, and furthermore themselves intersect in a smooth fashion. The following Lemma 4.2, which is constructive tells us how to go about this.

Lemma 4.3 *Let* $C_1 : (f_1 = 0, q_1 = 0)$ *and* $C_2 : (f_2 = 0, q_2 = 0)$ *be two conics in space. These two curves can be smoothly contained by two "smoothly intersecting" quadrics* $S_1 : g_1 = a_1 f_1 + b_1 q_1^2 = 0$ *and* $S_2 : g_2 = a_2 f_2 + b_2 q_2^2$ *if and only if there exist nonzero constants* a_1, a_2, b_1, b_2, α, β, *and a plane* $Q : q(x, y, z) = 0$ *such that* $a_1 f_1 + b_1 q_1^2 - \alpha(a_2 f_2 + b_2 q_2^2) - \beta q^2 = 0$.

Proof: From theorem 4.3 we note that two quadrics that intersect smoothly (at least C^1), must intersect with multiplicity at least two. It follows then from Bezout's theorem 4.1 for surface intersection, that the two quadrics S_1 and S_2 must meet in a plane curve (either an irreducible conic or straight lines). Let the intersection curve lie on the unknown plane Q, then just apply Lemma 4.1 three times. ♠

The final equation of the above Lemma results in a nonlinear (cubic) system of equations which is linear in terms of the unknowns a_1, a_2, b_1, b_2, α, and β, and quadratic in terms of the unknown coefficients of the plane $Q : q = 0$. Note, that in Lemma 4.3, the quadric surfaces S_1 and S_2 need not be in the form given (as constructed via Lemma 4.1), but may instead be an m-parameter family of solutions, obtained by C^1 interpolation of input curves with possibly "normal" data, as explained in the prior sections.

The above method of Lemma 4.3 can straightforwardly be extended to finding a C^1 continuous mesh of k quadric surfaces which smoothly contain k conics in space.

Theorem 4.4 *Let* $C_1 : (f_1 = 0, q_1 = 0)$, $C_2 : (f_2 = 0, q_2 = 0)$ $\ldots$ $C_k : (f_k = 0, q_k = 0)$ *be* k *conics in space. These curves can be smoothly contained by* k *quadrics* $S_1 : g_1 = a_1 f_1 + b_1 q_1^2 = 0$, $S_2 : g_2 = a_2 f_2 + b_2 q_2^2$, $\ldots$, $S_k : g_k = a_k f_k + b_k q_2^k$ *which themselves "smoothly intersect" if and only if* *there exist nonzero constants* $a_1, a_2, \ldots, a_k, b_1, b_2, \ldots, b_k, \alpha_1, \ldots, \alpha_{k-1}, \beta_1, \ldots, \beta_{k-1}$ *and planes* $R_1 : r_1(x,y,z) = 0, \ldots, R_{k-1} : r_{k-1}(x,y,z) = 0$ *such that*

$$\begin{aligned} a_1 f_1 + b_1 q_1^2 - \alpha_1(a_2 f_2 + b_2 q_2^2) - \beta_1 r_1^2 &= 0 \\ a_2 f_2 + b_2 q_2^2 - \alpha_2(a_3 f_3 + b_3 q_3^2) - \beta_2 r_2^2 &= 0 \\ &\ldots \\ a_{k-1} f_{k-1} + b_{k-1} q_{k-1}^2 - \alpha_{k-1}(a_k f_k + b_k q_k^2) - \beta_{k-1} r_{k-1}^2 &= 0 \end{aligned} \tag{2}$$

Proof: Direct applications of Lemma 4.3 ♠

Note again as before, that in the above theorem, the quadric surfaces $S_1, \ldots S_k$ need not be in the form given (as constructed via Lemma 4.1), but may instead be an m parameter family of solutions, obtained by C^1 interpolation of input curves with possibly "normal" data, as explained in the previous section. Also note, that given k conics in space, in general, k quadrics above, may not form a C^1 continuous mesh (no non-trivial solution for the generated system (2) of polynomial equations). In this case one may try increasing the number of quadric surface patches between any two of the given curves. This yields the theorem below, a variation of theorem 4.4.

Theorem 4.5 *Let* $C_1 : (f_1 = 0, q_1 = 0)$, *and* $C_2 : (f_2 = 0, q_2 = 0)$ *be two conics in space. These curves can be smoothly contained by two quadrics* $S_1 : g_1 = a_1 f_1 + b_1 q_1^2 = 0$, $S_2 : g_2 = a_2 f_2 + b_2 q_2^2$ *which together with* k *other quadrics* $T_1 : h_1 = 0, \ldots, T_k : h_k = 0$ *form a* C^1 *continuous mesh if and only if there exist nonzero constants* $a_1, a_2, b_1, b_2, c_{i0}, \ldots c_{i9}$ *(the coefficients of the quadric* $T_i : h_i = 0$*),* $i = 1 \ldots k$, *and* $\alpha_1, \ldots, \alpha_{k+1}, \beta_1, \ldots, \beta_{k+1}$, *and planes* $R_1 : r_1(x,y,z) = 0, \ldots, R_{k+1} : r_{k+1}(x,y,z) = 0$ *such that*

$$\begin{aligned} a_1 f_1 + b_1 q_1^2 - \alpha_1 h_1 - \beta_1 r_1^2 &= 0 \\ a_2 f_2 + b_2 q_2^2 - \alpha_{k+1} h_k - \beta_{k+1} r_{k+1}^2 &= 0 \\ h_i = \alpha_i h_{i-1} + \beta_i r_i^2, \quad i &= 2, \ldots, k \end{aligned} \tag{3}$$

.

Necessarily the complexity of the nonlinear system of equations also goes up.

If the above generated systems (2),(3) of polynomial equations, do not yield a satisfactory C^1 solution, one may instead try intermixing cubic surfaces with quadrics. To do this one first considers the lemma below similar to lemma 4.1 and a corollary of theorem 4.2

Lemma 4.4 *Let* $S : f(x,y,z) = 0$ *be an irreducible quadric surface, and* $Q : q(x,y,z) = 0$ *be a plane which intersects* S *in a conic* C. *Then, a cubic surface* $T_1 : f_1(x,y,z)$ *is tangent to* S *along* C *if and only if there exists nonzero constants* $a_1, \ldots, a_4$, *and* $b_1, \ldots, b_4$ *such that* $f_1 = (a_1 x + a_2 y + a_3 z + a_4) f + (b_1 x + b_2 y + b_3 z + b_4) q^2$.

Similar to lemma 4.3 one obtains

Lemma 4.5 *Let* $C_1 : (f_1 = 0, q_1 = 0)$ *and* $C_2 : (f_2 = 0, q_2 = 0)$ *be two conics in space. These two curves can be smoothly contained by two quadrics* $S_1 : g_1 = a_1 f_1 + b_1 q_1^2 = 0$ *and* $S_2 : g_2 =$

$a_2 f_2 + b_2 q_2^2$ both of which meet a cubic surface $T_1 : h_1 = 0$ if there exist nonzero constants a_1, a_2, b_1, b_2, $\alpha_{11}, \ldots, \alpha_{14}$, $\alpha_{21}, \ldots, \alpha_{24}$ $\beta_{11}, \ldots, \beta_{14}$, $\beta_{21}, \ldots, \beta_{24}$ and planes $R_1 : r_1(x,y,z) = 0$, $R_2 : r_2(x,y,z) = 0$ such that $h_1 = (\alpha_{11}x + \alpha_{12}y + \alpha_{13}z + \alpha_{14})g_1 + (\beta_{11}x + \beta_{12}y + \beta_{13}z + \beta_{14})r_1^2 = (\alpha_{21}x + \alpha_{22}y + \alpha_{23}z + \alpha_{24})g_2 - (\beta_{21}x + \beta_{22}y + \beta_{23}z + \beta_{24})r_2^2$

Proof: It follows from Bezout's theorem 4.1 for surface intersection, that the a quadrics S_1 and a cubic surface T_1 must meet in either a space cubic, a plane cubic, an irreducible conic or straight lines. Consider only the plane intersection curves and assume they lie on an unknown plane Q, then just apply Lemma 4.4 . ♠

In both the above lemmas, T_1 need not be in the above form but may instead be a l-parameter family of solutions, obtained by C^1 interpolation of input curves with possibly "normal" data, as explained in the previous section. These parameterized cubic surfaces may be intermixed with the quadric surfaces in theorems 4.4 and 4.5 to form a C^1 continuous mesh of alternating quadric and cubic surfaces in the obvious manner.

Examples

Ex 4.2 *A C^1 mesh of a family of Quadric and Quartic Surfaces*

Consider a wireframe of a solid model consisting of two circles, $C_1 : ((x^2 + y^2 + z^2 - 25 = 0, x = 0)$, and $C_2 : ((x^2 + y^2 + z^2 - 25 = 0, y = 0)$. Each curve is associated with a "normal" direction which is chosen in the same direction as the gradients of the sphere. That is, $n_1(x,y,z) = (0, 2y, 2z)$, and $n_2(x,y,z) = (2x, 0, 2z)$. The wireframe has 4 faces to be fleshed, $face_1 = (x \geq 0, y \geq 0)$, $face_2 = (x \geq 0, y \leq 0)$, $face_3 = (x \leq 0, y \leq 0)$, and $face_4 = (x \leq 0, y \geq 0)$.

In Figure 8z, $face_1$ and $face_3$ are filled with the patches taken from the sphere $x^2+y^2+z^2-25 = 0$ (yellow patches). To flesh the remaining faces with overall C^1 continuity along all inter-patch boundary curves, requires degree 4 surface patches. using the interpolation algorithms of section 2 yields C_1 and C_2, both 11-parameter (homogeneous) family of quartic G^1 interpolating surfaces, given by $f(x,y,z) = r_1 z^4 + (r_2 y + r_6 x + 5r_4)z^3 + (r_3 y^2 + (r_7 x + 5r_8)y + r_{10}x^2 + 5r_{11}x - 25r_9 - 25r_1)z^2 + (r_2 y^3 + (r_6 x + 5r_4)y^2 + (r_2 x^2 - 25r_2)y + r_6 x^3 + 5r_4 x^2 - 25r_6 x - 125r_4)z + (r_3 - r_1)y^4 + (r_7 x + 5r_8)y^3 + (r_5 x^2 + 5r_{11}x - 25r_9 - 25r_3 + 25r_1)y^2 + (r_7 x^3 + 5r_8 x^2 - 25r_7 x - 125r_8)y + (r_{10} - r_1)x^4 + 5r_{11}x^3 + (-25r_9 - 25r_{10} + 25r_1)x^2 - 125r_{11}x + 625r_9$. An instance from this family is $f(x,y,z) = -1250 - x^4 - y^4 - x^2z^2 - y^2z^2 + 50z^2 + 75y^2 + 75x^2$ used to fill faces $face_2$ and $face_4$ in Figure 8 (red patches). □

Ex 4.3 *A C^1 Mesh of Quadric Patches*

Let conic C_1 be given by $f_1 = x^2 + y^2 - z^2 + 4xy + 4x + 4y + 3 = 0$ (a hyperboloid of one sheet) and $q_1 = x + y + 1 = 0$. Similarly, let conic C_2 be given by $f_2 = 19x^2 + 10y^2 - 9z^2 + 38xy - 114x - 114y + 180 = 0$ (a hyperbooid of one sheet). $q_2 = x + y - 3 = 0$. and let the unknown plane be $P : ax + by + cz + d = 0$. Then the equation for the system of smooth interpolating quadrics $a_1 f_1 + b_1 q_1^2 - \alpha(a_2 f_2 + b_2 q_2^2) = \beta(ax + by + cz + d)^2$ results in a nonlinear system of 10 equations: $-\beta c^2 + 9a_2\alpha - a_1 = 0$, $-2b\beta c = 0$, $-2a\beta c = 0$, $-2\beta cd = 0$, $-b^2\beta - \alpha b_2 + b_1 - 10a_2\alpha + a_1 = 0$, $-2ab\beta - 2\alpha b_2 + 2b_1 - 38a_2\alpha + 4a_1 = 0$, $-2b\beta d + 6\alpha b_2 + 2b_1 + 114a_2\alpha + 4a_1 = 0$, $-a^2\beta - \alpha b_2 + b_1 -$

$19a_2\alpha + a_1 = 0$, $-2a\beta d + 6\alpha b_2 + 2b_1 + 114a_2\alpha + 4a_1 = 0$, and $-\beta d^2 - 9\alpha b_2 + b_1 - 180a_2\alpha + 3a_1 = 0$. This nonlinear system has a nontrivial solution (in the sense that a_1, a_2, and α are nonzero) : $a_1 = -a^2\beta$, $b_1 = 2a^2\beta$, $a_2 = -\frac{a^2\beta}{9\alpha}$, $b_2 = \frac{19a^2\beta}{9\alpha}$, and $b = c = d = 0$.[5] Hence, the two conics C_1 and C_2 are smoothly contained by quadrics $g_1 = 0$ and $g_2 = 0$, respectively, and which in turn, smoothly intersect in a conic in the plane Q. The real quadric $g_1 = x^2 + y^2 + z^2 - 1 = 0$ is a sphere, while the other real quadric $g_2 = y^2 + z^2 - 1$ is a cylinder. Note that the above solution implies that there is only one pair of real quadric surfaces which smoothly contain the given conics. Also, for this case, it can be shown that neither a single quadric nor a single cubic surface can Hermite interpolate the two given conics. Geometrically then, the two hyperboloids of one sheet are smoothly joined by a sphere and a cylinder. See figure 9.

Open Problems

1. Extend the technique of constructing C^1 continuous meshes to constructing C^k continuous meshes using the definition of C^k continuity of section 2 and the theorems of the above section.

5 Triangulated Data Fit with Surface Patches

Problem

Given a collection of Z-values and derivatives over a triangulation $\mathcal{T}$, construct a mesh of real algebraic surface patches S_i, which C^k interpolates the collection of points $\mathbf{p}_i = (x_i, y_i, z_i)$ in $\mathbb{R}^3$ and C^l least squares approximates the collection of points $\mathbf{q}_j = (x_j, y_j, z_j)$ in $\mathbb{R}^3$. The set of points $\mathbf{p}_i$ and $\mathbf{q}_i$ are not necessarily disjoint and $\mathcal{T}$ may be either an $X - Y$ or an $X - Y - Z$ triangulation of the entire collection of points.

Summary of Approaches

The generation of a mesh of smooth surface patches or *splines* that interpolate or approximate *triangulated space data* is one of the central topics of geometric design. Chui [23] and DeBoor [28] summarize much of the history of previous work. Interpolatory spline problems can be classified by the following factors :

- What kind of surface patches do they use for a basis : parametric, functional, or implicit?
- What kind of triangulation of data do they assume : functional values over 2D triangulation or arbitrarily spaced 3D triangulation?
- What kinds of information do they need as input data : C^0 data, C^1 data, C^2 data, and so on?
- With what order of geometric continuity do the patches meet along the boundary curves?
- Are they local, that is, each patch is constructed only from nearby data, or global?

[5] This nonlinear system was solved with the aid of MACSYMA, on a Symbolics 3650

- Do they split one macro triangle into many micro triangles or not?
- Do they handle general data or just special data?
- How ill-conditioned is the solution to perturbations in the input data ?
- How efficient are the algorithms?
- And many more ...

These splines are traditionally defined over a given planar triangulation with a polynomial function or parametric surface for each triangular face [3, 4, 16, 18, 37, 39, 47, 63]. Little work has been done on spline basis for implictly defined algebraic surfaces. Sederberg [60] shows how various smooth implicit surfaces can be manipulated as functions in Bezier control tetrahedra with finite weights. Dahmen [26] presents the construction of tangent plane continuous piecewise triangular quadric surfaces. In his construction a macro patch is split into 6 micro quadratic patches, similar to the splitting scheme of Powell-Sabin [53]. The resulting surfaces interpolate finite sets of essentially arbitrary points in $\mathbf{R}^3$ according to a given topology and given normal directions at the points within some ranges depending on the topology and the location of data points. Bajaj and Ihm [13] use a single implicit surface for each macro patch at the expense of a higher degree 5 surface. This quintic surface provides sufficient flexibility in globally C^1 surface fitting as well as provides local shape control.

Recent Results

The method of [13] takes as input a 3D triangulation $\mathcal{T}$ of points v_i in $\mathbb{R}^3$ with possibly first order derivative information ("normals") at the points. From the input, a wireframe mesh of conic curves is first constructed. Each conic replaces an edge e_j of the triangulation $\mathcal{T}$ and C^1 interpolates the end-points and specified normals at those points. If no normals are specified, a unique normal is chosen, for e.g. by averaging the normals of the incident faces f_k of $\mathcal{T}$ at that point. Unique normals, or in other words, unique tangent planes at the end-points is a necessary conditon for global C^1 fitting. Each conic being a rational curve, is represented by its rational parameterization $E_j(t) = (\frac{x(t)}{w(t)}, \frac{y(t)}{w(t)}, \frac{z(t)}{w(t)})$. Next, along each such conic edge a quadratically varying "normal" $N_j(t) = (\frac{nx(t)}{w(t)}, \frac{ny(t)}{w(t)}, \frac{nz(t)}{w(t)})$ is specified, with the property that $E_j'(t) \cdot N_j(t)$ is identically zero. The $N_j(t)$ necessarily take on the same values as the unique normals at the end-points.

Lee [44] presents a compact method for computing the above rational quadratic parametric representation for each of the edge conics. In particular, a ρ-conic parameterization is derived in which the remaining one degree of freedom of each conic edge is controlled in terms of the ρ value. The resulting topology of the wireframe is that of the initial triangulation $\mathcal{T}$, with each triangular face F_k now consisting of conic edges. The shape of each conic edge in turn being controlled by an independent ρ parameter.

Bajaj and Ihm [13] provide a local C^1 implicit surface interpolant S_k of degree 5 which fills each curvilinear triangular facet F_k. The degree 5 implicit surface suffices because its degrees of freedom are 55 while the maximum number of C^1 interpolatory constraints of F_k is 51. Each interpolant S_k has thus four independent parameters of local freedom, which are used for shape control. Further details on interactive shape control are in section 6.

Examples

Ex 5.1 *Locally supported triangular C^1 interpolants for smoothing polyhedra*

The input is an octahedron represented by vertices $v_i, i = 0...5$, edges $e_j, j = 0...11$ and faces $f_k, k = 0...7$, together with vertex normals $n_l, l = 0..5$.

$$\begin{aligned} v_0 &= (0.0, 0.0, 0.0) & v_1 &= (2.0, 4.0, 0.0) \\ v_2 &= (-0.1, 4.0, 2.1) & v_3 &= (2.0, 3.0, 2.0) \\ v_4 &= (1.0, 0.0, 2.0) & v_5 &= (2.5, 0.0, 1.0) \end{aligned}$$

$$\begin{aligned} e_0 &= (v_4, v_0) & e_1 &= (v_5, v_4) \\ e_2 &= (v_0, v_5) & e_3 &= (v_5, v_1) \\ e_4 &= (v_3, v_5) & e_5 &= (v_1, v_3) \\ e_6 &= (v_3, v_4) & e_7 &= (v_2, v_4) \\ e_8 &= (v_3, v_2) & e_9 &= (v_0, v_1) \\ e_{10} &= (v_1, v_2) & e_{11} &= (v_2, v_0) \end{aligned}$$

$$\begin{aligned} f_0 &= (e_0, e_4, e_5) & f_1 &= (e_1, e_5, e_3) \\ f_2 &= (e_5, e_4, e_3) & f_3 &= (e_4, e_2, e_3) \\ f_4 &= (e_1, e_0, e_5) & f_5 &= (e_1, e_3, e_2) \\ f_6 &= (e_2, e_4, e_0) & f_7 &= (e_2, e_0, e_1) \end{aligned}$$

$$\begin{aligned} n_0 &= (-0.592524, -0.557271, -0.581691) \\ n_1 &= (0.573733, 0.581132, -0.577162) \\ n_2 &= (-0.593023, 0.485480, 0.642365) \\ n_3 &= (0.633794, 0.243658, 0.734122) \\ n_4 &= (-0.104040, -0.537266, 0.836971) \\ n_5 &= (0.840500, -0.541705, -0.010696) \end{aligned}$$

First a wireframe of conics is constructed where each conic replaces an edge and C^1 interpolates the corresponding vertices of the edge. Next normals are constructed for each curvilinear conic edge of the wireframe and varying quadratically along the conics. Since the normals are quadratic functions and take on the value of the given normals at the vertex corners, specifying an additional normal vector at an interior point of each edge suffices.

$$edgenorm_0 = (-0.706837, 0.612762, 0.353418)$$

$$
\begin{aligned}
edgenorm_1 &= (0.426401, 0.639602, 0.639602) \\
edgenorm_2 &= (0.285582, 0.639307, -0.713954) \\
edgenorm_3 &= (-0.515202, -0.267929, -0.814114) \\
edgenorm_4 &= (-0.477035, -0.348458, 0.806855) \\
edgenorm_5 &= (-0.608621, 0.709693, 0.354847) \\
edgenorm_6 &= (-0.898131, 0.299377, 0.322078) \\
edgenorm_7 &= (-0.828637, -0.214953, -0.516872) \\
edgenorm_8 &= (0.339020, 0.766491, -0.545488) \\
edgenorm_9 &= (-0.874728, 0.437364, 0.208721) \\
edgenorm_{10} &= (-0.673887, -0.302905, -0.673887) \\
edgenorm_{11} &= (0.158349, 0.462070, -0.872592)
\end{aligned}
$$

See Figure 10(a) where the conics are constructed for the ρ value of 0.7.

The interpolation algorithm of the above section then constructs triangular C^1 interpolants - a 4 parameter family of quintic surfaces, one family per triangular facet of the wireframe. From these families a single instance is chosen for each facet. These are listed below and displayed in Figure 10(b).

$$
\begin{aligned}
facet_0(x,y,z) : &-0.000000000003 - 0.000000010437 * z - 0.351866100554 * z^2 \\
&+0.445697927749 * z^3 - 0.149228349853 * z^4 + 0.006409705518 * z^5 \\
&-0.000000010003 * y + 0.974854816609 * y * z + 1.047244055300 * y * z^2 \\
&-0.813301645559 * y * z^3 - 0.013779865829 * y * z^4 + 1.256871042976 * y^2 * z \\
&+4.799466010280 * y^2 * z - 1.773488566286 * y^2 * z^2 - 0.195440121483 * y^2 * z^3 \\
&+3.855336605660 * y^3 + 0.908723176875 * y^3 * z - 0.442893510064 * y^3 * z^2 \\
&+1.647368998297 * y^4 - 0.010965274559 * y^4 * z + 0.194339210883 * y^5 \\
&-0.000000010651 * x - 0.770526180505 * x * z + 0.804811773765 * x * z^2 \\
&-0.321648406719 * x * z^3 + 0.059462557369 * x * z^4 + 0.941573772622 * x * y \\
&+2.046806343664 * x * y * z - 1.745493230117 * x * y * z^2 + 0.282805348419 * x * y * z^3 \\
&+5.130053057327 * x * y^2 * z - 2.280499794089 * x * y^2 * z + 0.402577042726 * x * y^2 * z^2 \\
&+1.476015162086 * x * y^3 - 0.224321757547 * x * y^3 * z + 0.020644414286 * x * y^4 \\
&-0.419781907345 * x^2 + 0.707388836219 * x^2 * z - 0.361120269463 * x^2 * z^2 \\
&+0.052301598348 * x^2 * z^3 + 1.289787267161 * x^2 * y - 1.547823381213 * x^2 * y * z \\
&+0.548579135713 * x^2 * y * z^2 - 0.947015472962 * x^2 * y^2 + 0.344869768680 * x^2 * y^2 * z \\
&-0.458349727905 * x^2 * y^3 + 0.355415780406 * x^3 - 0.263713673860 * x^3 * z \\
&+0.040384255615 * x^3 * z^2 - 0.554279391319 * x^3 * y
\end{aligned}
$$

$$+0.244614404766*x^3*y*z-0.283785418206*x^3*y^2-0.072402573908*x^4$$
$$+0.041527902675*x^4*z-0.050857636030*x^4*y-0.001090526006*x^5$$

$$facet_1(x,y,z): -5.391392848589+1.655691554529*z-0.10054013$$
$$+0.010695554310*z^4+0.000015445336*z^5+0.391823657979*y-0.061907362326*$$
$$-0.018698359648*y*z^3+0.003550207987*y*z^4+0.301365510162$$
$$+0.022675047248*y^2*z^2-0.000878155318*y^2*z^3-0.062806190565*y^3+0.016993251409*y$$
$$+0.005728948099*y^4*-0.000835427317*y^4*z-0.0001971$$
$$-1.545576618926*x*z+0.012024864984*x*z^2+0.030745758603*$$
$$-0.641646581928*x*y*0.185959839052*x*y*z-0.046653590100*x*y*z^2+0.004901215538*x*y*$$
$$+0.043334490837*x*y^2*z-0.003159988252*x*y^2*z^2+0.020586984770*x*y$$
$$-0.000780290446*x*y^4*-3.648649986273*x^2+0.508225200029*x$$
$$-0.004480580992*x^2*z^3+0.327596884178*x^2*y-0.068505682280*x^2*y*$$
$$+0.034436178464*x^2*y^2*-0.004370304569*x^2*y^2*z-0.002134126507$$
$$-0.070554872584*x^3*z-0.004678977249*x^3*z^2-0.069717197288*x^3*$$
$$-0.002095499230*x^3*y^2*-0.108632430466*x^4+0.003372556775*x^4*z+0.005137768147$$

$$facet_2(x,y,z): -2.705288056399+4.638014160856*z-2.7683$$
$$+0.646097331317*z^3-0.036933002640*z^4-0.002695925262*z^5-3.328$$
$$+4.218478276240*y*z-1.745249397975*y*z^2+0.268535875414*y*z^3-0.0137341$$
$$-1.282509102765*y^2*1.101278297196*y^2*z-0.288331641811*y^2*z^2+0.0263072$$
$$-0.137253125822*y^3*0.059273271544*y^3*z-0.005863970988*y^3*z^2+0.0088$$
$$-0.004595032313*y^4*z+0.000746755484*y^5+1.892316604675*x*-3.146735171698*x*z+1.7645240$$
$$-0.374869320718*x*z^3$$

$$facet_3(x,y,z): 3.838736319407-7.136639001525*z+4.693530834395*z^2-1.357307983821*z^3+0.1784$$
$$-0.008750076057*z^5+1.122726212656*y-0.924668988889*y*z+0.0514823$$
$$+0.043930044239*y*z^3-0.005331770119*y*z^4-0.238054875023*y^2*0.3579077$$
$$-0.087082462242*y^2*z^2+0.005495431255*y^2*z^3-0.031650230022*y^3*-0.011879$$
$$+0.002685939576*y^3*z^2+0.005579405336*y^4-0.000368482605*y^4*z-0.0001$$
$$+1.976620035651*x*-1.835893595748*x*z+0.466818397131*x*z^2-0.0415045$$
$$+0.001006188013*x*z^4-0.078144287132*x*y+0.348845175392*x*y*z-0.0689316472$$
$$+0.001826387512*x*y*z^3-0.148628002020*x*y^2-0.025239912446*x*y^2*z+0.00703285133$$
$$+0.034316738617*x*y^3*-0.002423275496*x*y^3*z-0.001635387143*x*y^4*-0.36435$$

$$+0.506254734647 * x^2 * z - 0.163643901943 * x^2 * z^2 + 0.015748598439 * x^2 * z^3 - 0.16336189$$
$$+0.041338782134 * x^2 * y * z - 0.004218519660 * x^2 * y * z^2 + 0.033117806359 * x^2 * y^2 * -0.00171972149$$
$$-0.002952667841 * x^2 * y^3 * -0.117405799359 * x^3 + 0.044891233242 * x^3 * z - 0.00124717$$
$$-0.000490738665 * x^3 * y * -0.005187314917 * x^3 * y * z + 0.001670820372 * x^3 * y^2 * 0.01267$$
$$-0.006535458049 * x^4 * z + 0.004543426917 * x^4 * y + 0.00081$$

$$facet_4(x,y,z) : 0.000000000199 - 0.000000002731 * z + 0.011311493141 * z^2 - 0.707079497272 * z^3 - 0.09527$$
$$+0.000556875028 * z^5 - 0.000000002837 * y + 0.147337002989 * y * z - 1.97374861$$
$$-0.325401967187 * y * z^3 - 0.020075131188 * y * z^4 + 0.130769860685 * y^2 * -2.06141412$$
$$-0.399636480206 * y^2 * z^2 - 0.037686868985 * y^2 * z^3 - 0.782483149860 * y^3 * -0.21470272$$
$$-0.000404957002 * y^3 * z^2 - 0.043014811932 * y^4 + 0.046594653662 * y^4 * z + 0.02950$$
$$-0.000000003282 * x * 4.263562802083 * x * z - 0.734345998413 * x * z^2 + 0.22394520$$
$$+0.037962078216 * x * z^4 + 4.212574861013 * x * y + 0.157459011664 * x * y * z + 0.99206507873$$
$$+0.045649837633 * x * y * z^3 + 0.405290985255 * x * y^2 + 1.058072427816 * x * y^2 * z + 0.08378537692$$
$$+0.249319204394 * x * y^3 * 0.019810557415 * x * y^3 * z - 0.029459124470 * x * y^4 * 4.33122$$
$$-3.536565133014 * x^2 * z + 0.195350676910 * x^2 * z^2 + 0.028619664589 * x^2 * z^3 - 1.94969686$$
$$+0.339561326894 * x^2 * y * z - 0.181985409388 * x^2 * y * z^2 - 0.121048123933 * x^2 * y^2 * -0.06351114610$$
$$-0.041199050954 * x^2 * y^3 * -3.792880753729 * x^3 + 0.818135003255 * x^3 * z + 0.02992250$$
$$+0.260960871914 * x^3 * y * -0.164550553490 * x^3 * y * z + 0.065133155407 * x^3 * y^2 * 1.02734$$
$$-0.029769692122 * x^4 * z - 0.042320347953 * x^4 * y - 0.08200$$

$$facet_5(x,y,z) : -7.119827421214 + 2.494705342797 * z - 0.025897396309 * z^2 - 0.069015620217 * z^3 + 0.009$$
$$-0.000579660067 * z^5 + 5.119228344925 * y - 1.514263796801 * y * z + 0.021228$$
$$+0.018426213379 * y * z^3 - 0.000796939990 * y * z^4 - 1.438592661126 * y^2 * 0.345471$$
$$-0.005249912206 * y^2 * z^2 - 0.001456966611 * y^2 * z^3 + 0.196047588059 * y^3 * -0.034963$$
$$+0.000452558108 * y^3 * z^2 - 0.012821958380 * y^4 + 0.001312348329 * y^4 * z + 0.000$$
$$+2.837355448195 * x * -1.092599315171 * x * z + 0.085301176633 * x * z^2 + 0.00685$$
$$-0.000328923757 * x * z^4 - 1.640103182542 * x * y + 0.478243910269 * x * y * z - 0.02864265$$
$$-0.001509769984 * x * y * z^3 + 0.352606876101 * x * y^2 - 0.067471611808 * x * y^2 * z + 0.0027620029$$
$$-0.033501697454 * x * y^3 * 0.002970051350 * x * y^3 * z + 0.001195381560 * x * y^4 * -0.10609$$
$$+0.152063238488 * x^2 * z - 0.019091571055 * x^2 * z^2 + 0.000523406532 * x^2 * z^3 + 0.043862$$
$$-0.043360845178 * x^2 * y * z + 0.002746090652 * x^2 * y * z^2 - 0.006681624191 * x^2 * y^2 * 0.0029108789$$
$$+0.000397383658 * x^2 * y^3 * -0.095363563333 * x^3 - 0.005434040837 * x^3 * z + 0.0014043$$
$$+0.027731372156 * x^3 * y * 0.000611512007 * x^3 * y * z - 0.001963336031 * x^3 * y^2 * 0.015$$
$$-0.000136453125 * x^4 * z - 0.002235563671 * x^4 * y - 0.000$$

$$facet_6(x,y,z): -0.000000000001 + 0.000000004180 * z - 3.808420989548 * z^2 + 4.493996112967 * z^3 - 1.611$$
$$+0.160134916841 * z^5 + 0.000000004005 * y - 2.922530506296 * y * z + 1.086707$$
$$+0.171045406740 * y * z^3 + 0.016939537720 * y * z^4 + 0.695527270492 * y^2 * -1.728567$$
$$+0.628183410140 * y^2 * z^2 - 0.156300642717 * y^2 * z^3 + 0.843291361707 * y^3 * -0.439536$$
$$+0.057573500599 * y^3 * z^2 + 0.073538735839 * y^4 + 0.050084868667 * y^4 * z - 0.032$$
$$+0.000000004185 * x * -3.167172786099 * x * z + 2.598846923454 * x * z^2 - 0.407822$$
$$-0.052377199454 * x * z^4 + 1.421797370428 * x * y - 3.335438470121 * x * y * z + 1.081415686$$
$$+0.043804038796 * x * y * z^3 + 2.114689484420 * x * y^2 - 1.504298946820 * x * y^2 * z + 0.1049082175$$
$$+0.355506220977 * x * y^3 * 0.000666711948 * x * y^3 * z - 0.043221139207 * x * y^4 * 0.725$$
$$-0.840946698144 * x^2 * z + 0.355870678528 * x^2 * z^2 - 0.065105570761 * x^2 * z^3 + 2.086369$$
$$-1.565502136001 * x^2 * y * z + 0.298582438735 * x^2 * y * z^2 + 0.624337656078 * x^2 * y^2 * -0.1818616232$$
$$+0.003740004555 * x^2 * y^3 * 0.857378484356 * x^3 - 0.518862065863 * x^3 * z + 0.0447867$$
$$+0.492724657128 * x^3 * y * -0.124509821514 * x^3 * y * z + 0.046350113632 * x^3 * y^2 * 0.143$$
$$-0.055879147512 * x^4 * z + 0.038417706086 * x^4 * y + 0.004$$

$$facet_7(x,y,z): 0.000000000156 - 0.000000294686 * z + 2.349287281326 * z^2 + 1.559485859400 * z^3 - 0.17182$$
$$-0.046419677846 * z^5 - 0.000000282296 * y - 2.190582451529 * y * z + 0.73366187$$
$$-0.464254769910 * y * z^3 - 0.019296336841 * y * z^4 - 4.254791369719 * y^2 * 0.98635737$$
$$-0.246306877928 * y^2 * z^2 + 0.046011762281 * y^2 * z^3 + 1.840122660302 * y^3 * -0.1692327$$
$$+0.013962126112 * y^3 * z^2 - 0.260578353453 * y^4 + 0.010138447483 * y^4 * z + 0.01212$$
$$-0.000000300007 * x * 4.667782885584 * x * z + 3.351301552208 * x * z^2 - 0.1588711$$
$$-0.005164712180 * x * z^4 - 2.344699354362 * x * y - 0.299486486776 * x * y * z - 1.10703932097$$
$$+0.081566524091 * x * y * z^3 + 0.929213069532 * x * y^2 - 0.525207873539 * x * y^2 * z + 0.0856528827$$
$$-0.107832884419 * x * y^3 * 0.065377127126 * x * y^3 * z + 0.003356701228 * x * y^4 * 2.31710$$
$$+3.490045918816 * x^2 * z + 0.034625547103 * x^2 * z^2 + 0.046251908196 * x^2 * z^3 + 0.73442340$$
$$-1.197735369621 * x^2 * y * z - 0.014366274280 * x^2 * y * z^2 - 0.248426455413 * x^2 * y^2 * 0.10044738535$$
$$+0.015222606449 * x^2 * y^3 * 1.720783062333 * x^3 - 0.078554062782 * x^3 * z + 0.05122518$$
$$-0.476963551788 * x^3 * y + 0.063955442264 * x^3 * y * z + 0.029515511569 * x^3 * y^2 - 0.10432$$
$$+0.004473642110 * x^4 * z - 0.039095324000 * x^4 * y - 0.04428$$

Open Problems

1. Build a quadratic wireframe for arbitrarily shaped 3D triangulated data.
2. Provide interactive shape control for a p parameter family of implicit surfaces.
3. Provide trade-off bounds of degree of implicit algebraic surfaces and the number of split patches in C^k interpolation.

6 Interactive Shape Control of Implicit Surface Patches

Problem

Interactively control the shape of an interpolating or approximating implicit surface by selecting appropriate instances from a p-parameter family of solution surfaces.

Summary of Approaches

The problem of interactively selecting a surface instance from a p-parameter family of solutions is equivalent to assigning values to each of the p parameters. When there are p parameters to be instantiated, one may additionally specify a set of points, curves or even surfaces around the earlier given input data, which approximately describes the final surface to be designed. The final solution instance is computed via interpolation of the given input data and with least-squares approximation of the additional data set. This scheme is presented in the paper by Bajaj, Ihm and Warren [14]. An example of this use is presented in Ex 4.1 of section 4.

An alternate scheme based on Sederberg's Bezier formulation of algebraic surfaces [60] is used for shape control of a family of surfaces in Bajaj and Ihm [12]. They present a method which allows a surface designer to intuitively and interactively control the shape of a C^k interpolating or approximating surface, thereby choosing an appropriate instance from the family by automatically selecting values for the p distinct parameters.

Recent Results

The result of multivariate C^k interpolation algorithm for a given data set, is in general, a p-parameter family of algebraic surfaces $f(x, y, z) = 0$, satisfying the given geometric data constraints. A surface designer must be able to choose an appropriate instance from this family, to satisfy his application by specifying values for the p parameters, (say $\mathbf{r} = r_1, r_2, \cdots, r_p$). The equation for the family has the form

$$f(x,y,z) = \sum_{i=0}^{n} \sum_{j=0}^{n-i} \sum_{k=0}^{n-i-j} c_{ijk} \cdot x^i y^j z^k = 0 \tag{4}$$

where c_{ijk} is a homogeneous linear combination of $\mathbf{r}$. Various distinct choices of values for $\mathbf{r}$ yields interpolating surface instances possessing different shapes. Bajaj and Ihm [12] present a method which allows a surface designer to intuitively and interactively control the shape of a Hermite interpolating surface, thereby choosing an appropriate instance from the family by automatically selecting values for the p distinct parameters.

The essential idea is to consider the interpolating family f as the zero contour $w = 0$ of the trivariate function $w = f(x, y, z)$. See Sederberg [60] where the same idea is used to define algebraic surface patches. Of course, since one considers a family of interpolating algebraic surfaces, the coefficients of f here have indeterminates r_i. The trivariate function, when transformed into Barycentric coordinates yields a control polyhedron with weights (the interactive control given to the designer). For the purpose here, the trivariate polynomial $f(x, y, z)$ is symbolically converted into a polynomial $F(s, t, u)$ in Barycentric coordinates, specified over a tetrahedron. To concentrate on a specific portion of the algebraic surface, the designer appropriately chooses the location of the four vertices of the tetrahedron, enclosing the desired region. Shape of the Hermite interpolating surface is now controlled by changing the weights of control points associated with the tetrahedron.

Let the trivariate Barycentric coordinates of points inside a tetrahedron be given by (s,t,u). The tetrahedron is specified by the designer who selects the location of its four vertices P_{n000}, P_{0n00}, P_{00n0}, and P_{000n}. The Cartesian coordinates P of a point inside the tetrahedron are related to its Barycentric coordinates s,t,u by $P = sP_{n000} + tP_{0n00} + uP_{00n0} + (1-s-t-u)P_{000n}$. Control points on the tetrahedron are defined by $P_{ijk} = \frac{i}{n}P_{n000} + \frac{j}{n}P_{0n00} + \frac{k}{n}P_{00n0} + \frac{n-i-j-k}{n}P_{000n}$ for nonnegative integers i,j,k such that $i+j+k \leq n$. With each control point there is also associated a weight w_{ijk}, corresponding to the coefficients c_{ijk} of (4), which is a linear (not necessarily homogeneous) combination of $\mathbf{r}$. All this together defines the p-parameter algebraic surface family in Barycentric coordinates,

$$F(s,t,u) = \sum_{i=0}^{n}\sum_{j=0}^{n-i}\sum_{k=0}^{n-i-j} w_{ijk} \cdot \binom{n}{i,j,k} \cdot s^i t^j u^k (1-s-t-u)^{n-i-j-k} = 0 \tag{5}$$

There are $\binom{n+3}{3}$ w_{ijk}, exactly as many as the c_{ijk}. Straightforward methods exist to converting a trivariate polynomial in the power basis with cartesian coordinates, to the form above in trivariate barycentric coordinates defined over a given tetrahedron.

Consider, as a simple example, a quadric surface which Hermite interpolates a line $LN : (1-t,t,0)$ with a normal $(0,0,1)$. Hermite interpolation algorithm returns a 5-parameter family of surfaces $f(x,y,z)$ as in (4) with $n=2$ and where $c_{200} = r_1$, $c_{110} = 2r_1$, $c_{101} = r_4$, $c_{100} = -2r_1$, $c_{020} = r_1$, $c_{011} = r_5$, $c_{010} = -2r_1$, $c_{002} = r_3$, $c_{001} = r2$,and $c_{000} = r_1$. For a given tetrahedron with vertices $P_{n00} = (2,0,0)$, $P_{0n0} = (0,2,0)$, $P_{00n} = (0,0,2)$, and $P_{000} = (0,0,0)$, the surface family representation $f(x,y,z)$ is transformed to $F(s,t,u)$ as in (5) with $n=2$ and where $w_{000} = r_1$, $w_{001} = r_1 + r_2$, $w_{002} = r_1 + 2r_2 + 4r_3$, $w_{010} = -r_1$, $w_{011} = -r_1 + r_2 + 2r_5$, $w_{020} = r_1$, $w_{100} = -r_1$, $w_{101} = -r_1 + r_2 + 2r_4$, $w_{110} = r_1$, and $w_{200} = r_1$.

As $F(s,t,u) = 0$ in (5), describes a constrained p-parameter family of algebraic surfaces of degree n, the change of one of the weights w_{ijk} associated with a control point of the tetrahedron, affects the weights of other control points. For example, suppose $w_1 = r_1 + r_2 + r_3 + 2r_4 - 1$, $w_2 = r_1 + r_2 + r_4 + 5$, and $w_3 = r_3 + r_4$. Then, we can derive the following linear relation between the weights, : $w_1 - w_2 - w3 - 6 = 0$. (For notational simplicity we here consider, wlg, the weights w_{ijk} to be also indexed by a single parameter l, i.e. weights w_l). From this invariant, we get $\Delta w_1 - \Delta w_2 - \Delta w_3 = 0$, and every time weights are changed, the above invariant is maintained.

Hence, in general one derives a system of invariant equations

$$\begin{aligned} I_1(\Delta w_1, \Delta w_2, \cdots, \Delta w_c) &= 0 \\ I_2(\Delta w_1, \Delta w_2, \cdots, \Delta w_c) &= 0 \\ &\vdots \\ I_l(\Delta w_1, \Delta w_2, \cdots, \Delta w_c) &= 0 \end{aligned}$$

from the linear weight expressions

$$w_1(r_1, r_2, \cdots, r_p) = w_1$$

$$\begin{aligned} w_2(r_1, r_2, \cdots, r_p) &= w_2 \\ &\vdots \\ w_c(r_1, r_2, \cdots, r_p) &= w_c \end{aligned}$$

This is easily achieved by Gaussian elimination. Changing some weights can now be considered as moving from a weight vector $\mathbf{W} = (w_1, w_2, \cdots, w_c)$ to another $\mathbf{W}' = (w_1', w_2', \cdots, w_c')$, with the constraint that $\Delta\mathbf{W} = \mathbf{W}' - \mathbf{W}$ is a solution of the computed system of invariant equations. Next, suppose that a surface designer wants to see how the surface shape changes with a value change of w_1 alone. However, a change in value of w_1 automatically changes the value of additional w_i's related to it by the invariants I_j's. Usually, the linear system of invariant equations are underdetermined, yielding an infinite number of choices of Δw_i's$(i = 2, 3, \cdots, c)$. Then how does the designer make a choice of values for the w_i's that reflects the influence of a change of only w_1, as clearly as possible?

One possible heuristic is to minimize the 2-norm of $(\Delta w_2, \cdots, \Delta w_c)$, and hence the 2-norm of $\Delta\mathbf{W}$. Note that $\|\Delta\mathbf{W}\|_2^2 = \Delta w_1^2 + \Delta w_2^2 + \cdots + \Delta w_c^2$. For a change $\Delta w_1 = d$, one sees that

$$\begin{aligned} I_1(d, \Delta w_2, \cdots, \Delta w_c) &= 0 \\ I_2(d, \Delta w_2, \cdots, \Delta w_c) &= 0 \\ &\vdots \\ I_l(d, \Delta w_2, \cdots, \Delta w_c) &= 0 \end{aligned}$$

will have a solution $\Delta\mathbf{W}^0 = (d, \Delta w_2^0, \cdots, \Delta w_c^0)$ where Δw_i^0's are expressed linearly through another set of free parameters $q_1, q_2, \cdots, q_{p-1}$. Hence, $\|\Delta\mathbf{W}^0\|_2$ is a quadratic function of the new parameters, which we denote by $Q(q_1, q_2, \cdots, q_{p-1})$.

In order to minimize the norm of $\Delta\mathbf{W}_0$, the quadratic function $Q(q_1, q_2, \cdots, q_{p-1})$ needs to be minimized. Since Q is quadratic, the minimum solution can be obtained straightforwardly by solving the linear system $\nabla Q(q_1, q_2, \cdots, q_{p-1}) = 0$. If the (unique) minimum solution point is $Q^0 = (q_1^0, q_2^0, \cdots, q_{(p-1)}^0)$, then $\Delta\mathbf{W}^0 = (d, \Delta w_2^0, \cdots, \Delta w_c^0)$ corresponding to Q^0 will define the desired change of weights of $w_2, \cdots, w_n$ having minimum effect on the shape of the surface. The instance surface for the new weights $\mathbf{W}' = \mathbf{W} + \Delta\mathbf{W}^0$ will then reflect predominantly the effect of the change of w_1 by $\Delta w_1 = d$.

Examples

Ex 6.1 *Interactive Shape Control of a Family of Quartic Surfaces*

Construct the lowest degree surface which can smoothly join three truncated orthogonal circular cylinders $CYL_1 : x^2+y^2-1 = 0$ for $z \geq 2$, $CYL_2 : y^2+z^2-1 = 0$ for $x \geq 2$, and $CYL_3 : z^2+x^2-1 = 0$ for $y \geq 2$. C^1 interpolation shows that the minimum degree for such a joining surface is 4, and the null space of the interpolation matrix is a 2 homogenous parameter (or one independent affine parameter) family of algebraic surfaces. Consider a circle $C_1 : (\frac{2t}{1+t^2}, \frac{1-t^2}{1+t^2}, 2)$ on CYL_1 with the associated rational "normal" $\mathbf{n}_1(t) : (\frac{4t}{1+t^2}, \frac{2-2t^2}{1+t^2}, 0)$, the circle $C_2 : (2, \frac{2t}{1+t^2}, \frac{1-t^2}{1+t^2})$ on CYL_2 with the associated rational "normal" $\mathbf{n}_2(t) : (0, \frac{4t}{1+t^2}, \frac{2-2t^2}{1+t^2})$, and the circle $C_3 : (\frac{2t}{1+t^2}, 2, \frac{1-t^2}{1+t^2})$ on CYL_3 with the associated rational "normal" $\mathbf{n}_3(t) : (\frac{4t}{1+t^2}, 2, \frac{2-2t^2}{1+t^2})$.

Again, all C_1, C_2 and C_3's "normals" are respectively chosen in the same direction as the gradients of their corresponding containing surfaces CYL_1, CYL_2, and CYL_3. This ensures that any interpolating surface for C_1, C_2, and C_3 will also meet CYL_1, CYL_2, and CYL_3 smoothly along these curves. A degree three algebraic surface does not suffice since the rank of the resulting linear system is greater than 19, the number of independent unknowns. Next as a possible interpolant one considers a degree four algebraic surface with 34 independent unknown coefficients. Repeating the interpolation for the original curves results in 52 equations. The rank of this linear system is 33, and thus there is a one affine independent parameter family of quartic Hermite interpolating surface, which is $f(x,y,z) = r_1 z^4 + \frac{r_2+10r_1}{12} yz^3 + \frac{r_2+10r_1}{12} xz^3 - \frac{r_2+10r_1}{3} z^3 + 2r_1 y^2 z^2 + \frac{r_2+10r_1}{12} xyz^2 - \frac{r_2+10r_1}{3} yz^2 + 2r_1 x^2 z^2 - \frac{r_2+10r_1}{3} xz^2 + r_2 z^2 + \frac{r_2+10r_1}{12} y^3 z + \frac{r_2+10r_1}{12} xy^2 z - \frac{r_2+10r_1}{3} y^2 z + \frac{r_2+10r_1}{12} x^2 yz - \frac{r_2+10r_1}{3} xyz + \frac{r_2+10r_1}{4} yz + \frac{r_2+10r_1}{12} x^3 z - \frac{r_2+10r_1}{3} x^2 z + \frac{r_2+10r_1}{4} xz + \frac{r_2+10r_1}{3} z + r_1 y^4 + \frac{r_2+10r_1}{12} xy^3 - \frac{r_2+10r_1}{3} y^3 + 2r_1 x^2 y^2 - \frac{r_2+10r_1}{3} xy^2 + r_2 y^2 + \frac{r_2+10r_1}{12} x^3 y - \frac{r_2+10r_1}{3} x^2 y + \frac{r_2+10r_1}{4} xy + \frac{r_2+10r_1}{3} y + r_1 x^4 - \frac{r_2+10r_1}{3} x^3 + r_2 x^2 + \frac{r_2+10r_1}{3} x + \frac{5r_1-7r_2}{3}$.

An instance of this family ($r_1 = 1$, $r_2 = 10$) is shown in Figure 11. Figure 11 illustrates three different instances of $f(x,y,z) = 0$ obtained by changing the value of w_{000}. Each time w_{000} is varied, the invariant equations are met, and each instance surface Hermite G^1 interpolates the three input curves. As the value of w_{000} continually increases from $w_{000} < 0$, the surface eventually passes through $P_{000} = (0,0,0)$ for $w_{000} = 0$, and then separates into 3 irreducible parts for $w_{000} > 0$. □

Open Problems

1. Derive other intuitive techniques for interactive shape control ?

7 Conclusion

Many of the interpolation and approximation algorithms presented here were implemented by Insung Ihm as part of SHILP, our solid modeling and display system [7]. The programs take as input any collection of geometric data points and curves, with and without associated "normals" and their derivatives. Both implicit and rational parametric representations of the space curves and their derivatives are allowed. The rank computation is done implicitly during the solution steps. The eigenvalue computation for least square computations is done with the help of routines from EISPACK. The result, when nontrivial solutions exist, are expressed in terms of symbolic coefficients and represent a family of interpolation surfaces. Values are specified for these coefficients by means of either the least-squares approximation approach as indicated in section 3, or using Bezier control weights as detailed in section 6. Desirable nonsingular and irreducible, real algebraic surfaces are computed. Insung Ihm is currently improving this implementation to include, a more user-friendly method of instantiating the interpolated solution, as well as a way of automatically incorporating the nonsingular and irreducibility constraints [40].

Acknowledgements I am grateful to my graduate student Insung Ihm for all his help.

References

[1] Albano, A., (1974) "Representation of Digitized Contours in Terms of Conic Arcs and Straight Line Segments", *Computer Graphics and Image Processing*, 3, 23 - 33.

[2] Alfeld, P., (1989) "Scattered Data Interpolation in Three or More Variables", *Mathematical Methods in Computer Aided Geometric Design*, ed. Lyche, T. and Schumaker, L., 1-34.

[3] Alfeld, P., Piper, B., and Schumaker, L., (1987) "An Explicit Basis for C^1 Quartic Bivariate Splines", *Siam Journal of Numerical Analysis*, 24, 891 - 911.

[4] Alfeld, P. Piper, B., and Schumaker, L., (1987) " Minimally Supported Bases for Spaces of Bivariate Piecewise Polynomials of Smoothness r for Degree $d > 4r + 1$", *Computer Aided Geometric Design*, 4, 105 - 123.

[5] Alfeld, P., Piper, B., and Schumaker, L., (1988) "Spaces of Bivariate Splines on Triangulations with Holes", *Journal of Approximation Theory*,

[6] Alfeld, P. and Schumaker, L., (1987) "The Dimension of Bivariate Spline Spaces of Smoothness r for Degree $d > 4r + 1$", *Constructive Approximation*, 3, 2, 189-197.

[7] Anupam, V., Bajaj, C., Dey, T., Ihm, I., Klinkner, S., (1990), "The SHILP Solid Model Creation, Editing and Display Toolkit", Computer Science Tech. Report, Purdue University, CSD-TR-988.

[8] Bajaj, C., (1988) "Geometric Modeling with Algebraic Surfaces" *The Mathematics of Surfaces III*, ed., D. Handscomb, Oxford University Press, 3 - 48.

[9] Bajaj, C., (1990) "C^1 Interpolation using Piecewise Quadrics and Cubic Surfaces", *Proc. of the Symposium on Electronic Imaging Science & Technology*, Santa Clara, CA.

[10] Bajaj, C., Dyksen, W., Hoffmann, C., Houstis, E., Korb, J., Rice, J., (1988) "Computing About Physical Objects" *Proc. of the 12th IMACS WORLD Congress*, Paris, 642-645.

[11] Bajaj, C. and Ihm, I., (1989), "Hermite Interpolation using Real Algebraic Surfaces", *Proc. of the Fifth ACM Symposium on Computational Geometry*, West Germany, 94-103.

[12] Bajaj, C. and Ihm, I., (1990), "Algebraic Surface Design with Hermite Interpolation" Computer Science Technical Report, TR-939, CAPO-90-1, Purdue University.

[13] Bajaj, C. and Ihm, I., (1990), "Polyhedral Smoothing using Implicit Patches with Local Control" Manuscript.

[14] Bajaj, C., Ihm, I., and Warren, J., (1990). "Exact and Least-Squares Approximate G^k Fitting of Implicit Algebraic Surfaces" Computer Science Technical Report, TR-944, CAPO-90-4, Purdue University.

[15] Biggerstaff, R., (1972), "Three Variations in Dental Arch Form Estimated by a Quadratic Equation", *Journal of Dental Research*, 51, 1509.

[16] Billera, L. (1988) "Homology of Smooth Splines: Generic Triangulations and a Conjecture of Strang", *Transactions of the American Mathematical Society*, 310, 1, 325 - 340.

[17] Billera, L. (1989) "The Algebra of Continuous Piecewise Polynomials over a Simplicial Complex" *Advances in Mathematics*, xxxx

[18] Billera, L. and Rose, L., (1989) "Grobner Basis Methods for Multivariate Splines", *Mathematical Methods in Computer Aided Geometric Design*, ed. Lyche, T. and Schumaker, L., 93-104

[19] Bookstein, F., L., (1979) "Fitting Conic Sections to Scattered Data", *Computer Graphics and Image Processing*, 9, 56-71.

[20] Bohm, W., Farin, G. and Kahmann, J., (1984), "A Survey of Curves and Surface Methods in CAGD", *Computer Aided Geometric Design*, 1, 1 - 60.

[21] Carnicer, J. and Gasca, M., (1989) "On Finite Element Interpolation Problems", *Mathematical Methods in Computer Aided Geometric Design*, ed. Lyche, T. and Schumaker, L., 105-114

[22] Cooper, D., and Yalbik, N., (1976), "On the Computational Cost of Approximating and Recognizing Noise Perturbed Straight Lines and Quadratic Arcs in the Plane", *IEEE Transactions on Computers*, C-25, 10, 1020-1032.

[23] Chui, C., (1988) *Multivariate Splines*, CBMS-NSF, Regional Conference Series in Applied Mathematics, 54, SIAM.

[24] Chui, C., and Wang, R., (1983) "On Smooth Multivariate Spline Functions", *Mathematics of Computation*, 41, 131 - 142.

[25] Chui, C., and Wang, R., (1983) "Multivariate Spline Spaces", *Journal of Mathematical Analysis and Applications*, 94, 197 - 221.

[26] Dahmen, W., (1989) "Smooth Piecewise Quadric Surfaces", *Mathematical Methods in Computer Aided Geometric Design*, ed. Lyche, T. and Schumaker, L., 181-194

[27] Dahmen, W. and Micchelli, C., (1987) "Algebraic Properties of Discrete Box Splines", *Constructive Approximation*, vol. 3, no. 2, ed. Hollig, K., 209-222

[28] DeBoor, C., (1990), "Quasiinterpolants and Approximation Power of Multivariate Splines", *Computation of Curves and Surfaces*, ed. Dahmen, W., Gasca, M., and Michelli, C., 313-346.

[29] DeRose, T., (1985), "Geometric Continuity: A Parameterization Independent Measure of Continuity for Computer Aided Design", Ph.D. Thesis, Computer Science, University of California, Berkeley.

[30] Farin, G., (1986), "Triangular Berstein-Bezier Patches", *Computer Aided Geometric Design*, 3, 83 - 127.

[31] Fjallstrom, P., (1986), "Smoothing of Polyhedral Models", *Proc. of the Second ACM Symposium on Computational Geometry*, Yorktown Heights, NY, 226-235.

[32] Franke, D., (1987) "Survey on Data Fitting",

[33] Fulton, W., (1984), *Intersection Theory*, Springer Verlag.

[34] Garrity, T., and Warren, J., (1989), "Geometric Continuity", Computer Science Technical Report TR88-89, Rice University.

[35] Gnanadesikan, R., (1977), *Methods for Statistical Data Analysis of Multivariate Observations*, John Wiley.

[36] Golub, G., and Underwood, R., (1970), "Stationary Values of the Ratio of Quadratic Forms Subject to Linear Constraints", *Z. Agnew. Math. Phys.*, 21, 1970, 318-326.

[37] Hagen, H. and Pottmann, H., (1989) "Curvature Continuous Triangular Interpolants", *Mathematical Methods in Computer Aided Geometric Design*, ed. Lyche, T. and Schumaker, L., 373-384

[38] Hoffmann, C., and Hopcroft, J., (1986), "Quadratic Blending Surfaces", *Computer Aided Design*, 18, 301-306.

[39] Hollig, K., (1989) "Box-Spline Surfaces", *Mathematical Methods in Computer Aided Geometric Design*, ed. Lyche, T. and Schumaker, L., 385-402

[40] Ihm, I. (1991) "Surface Design with Implicit Algebraic Surfaces" Ph.D. Thesis in preparation.

[41] Kendig, K., (1977) *Elementary Algebraic Geometry*, Graduate texts in Mathematics 44, Springer Verlag.

[42] Kunz, E., (1985) *Introduction to Commutative Algebra and Algebraic Geometry*, Birkhauser.

[43] Lawson, C., and Hanson, R., (1974). *Solving Least-Squares Problems*, Prentice Hall.

[44] Lee, E., (1987), "The Rational Bezier Representation for Conics", *Geometric Modelling:Algorithms and New Trends. G. Farin, editor, SIAM, Philadelphia*, 3 - 19.

[45] Lorentz, R., (1989) "Uniform Bivariate Hermite Interpolation", *Mathematical Methods in Computer Aided Geometric Design*, ed. Lyche. T. and Schumaker, L., 435-444

[46] Middleditch, A., and Sears, K., (1985), "Blend Surfaces for Set Theoretic Volume Modeling Systems", *Computer Graphics*. (Proc. Siggraph '85), 19, 161-170.

[47] Morgan, J., and Scott, R., (1975) "A Nodal Basis for C^1 piecewise polynomials of degree $n \geq 5$", *Mathematics of Computation*, 29, 736 - 740.

[48] Owen, J., and Rockwood, A., (1987), "Blending Surfaces in Solid Geometric Modeling ", *Geometric Modeling: Algorithms and New Trends*. ed. G. Farin, SIAM, 347-366.

[49] Paton, K., (1970), "Conic Sections in Chromosome Analysis", *Pattern Recognition*, 2, 39-51.

[50] Peters, J., (1989) "Smooth Interpolation of a Mesh of Curves" Univ of Wisconsin, Tech. Report.

[51] Peters, J., and Sitharam, M.,(1989) "Interpolation from C^1-cubics at the vertices of an underlying triangulation", Univ of Wisconsin, Tech. Report. #90-13.

[52] Pfluger, P. and Gmeling, R., (1989) "An Algorithm for Smooth Interpolation to Scattered Data in R^2 ", *Mathematical Methods in Computer Aided Geometric Design*, ed. Lyche, T. and Schumaker, L., 469-480

[53] Powell, M., and Sabin, M., (1977) "Piecewise Quadratic Approximations on Triangles" *ACM Trans. on Math. Software* 3, 316-325.

[54] Pratt, V., (1987), "Direct Least Squares Fitting of Algebraic Surfaces", *Computer Graphics*, (Proc. Siggraph '87), 21, 145-152.

[55] Rossignac, J., and Requicha, A., (1984), "Constant-Radius Blending in Solid Modeling", *Computers in Mathematical Engineering*, 2, 655-673.

[56] Sabin, M., (1989) "Open Questions in the Application of Multivariate B-splines" *Mathematical Methods in Computer Aided Geometric Design*, ed. Lyche, T. and Schumaker, L., 529-538

[57] Sarraga, R., (1987), "G^1 Interpolation of generally unrestricted Cubic Bezier Curves", *Computer Aided Geometric Design*, 4, 23-39.

[58] Sampson, P., (1982), "Fitting Conic Sections to Very Scattered Data: An Iterative Refinement of the Bookstein Algorithm", *Computer Graphics and Image Processing*, 18, 97-108.

[59] Schumaker, L., (1979) "On the Dimension of Spaces of Piecewise Polynomials in Two Variables", *Multivariate Approximation Theory*, eds. W. Schemp and K. Zeller, Birkhäuser, 396 - 412.

[60] Sederberg, T., (1985), "Piecewise Algebraic Surface Patches", *Computer Aided Geometric Design*, 2, 53-59.

[61] Strang, G., (1973) "Piecewise Polynomials and the Finite Element Method", *Bulletin of the American Mathematical Society*, 79, 1128 - 1137,

[62] Warren, J., (1986), "On Algebraic Surfaces Meeting with Geometric Continuity", Ph.D. Thesis, Computer Science, Cornell University, Ithaca, New York.

[63] Whitely, W., (1989) "A Matrix for Splines", *Journal of Approximation Theory*,

[64] Woodwark, J., (1987), "Blends in Geometric Modeling", *The Mathematics of Surfaces II*, ed. R. Martin, 255-297.

[65] Zariski, O. and Samuel, P. (1958), *Commutative Algebra (Vol.I, II)*, Springer Verlag.

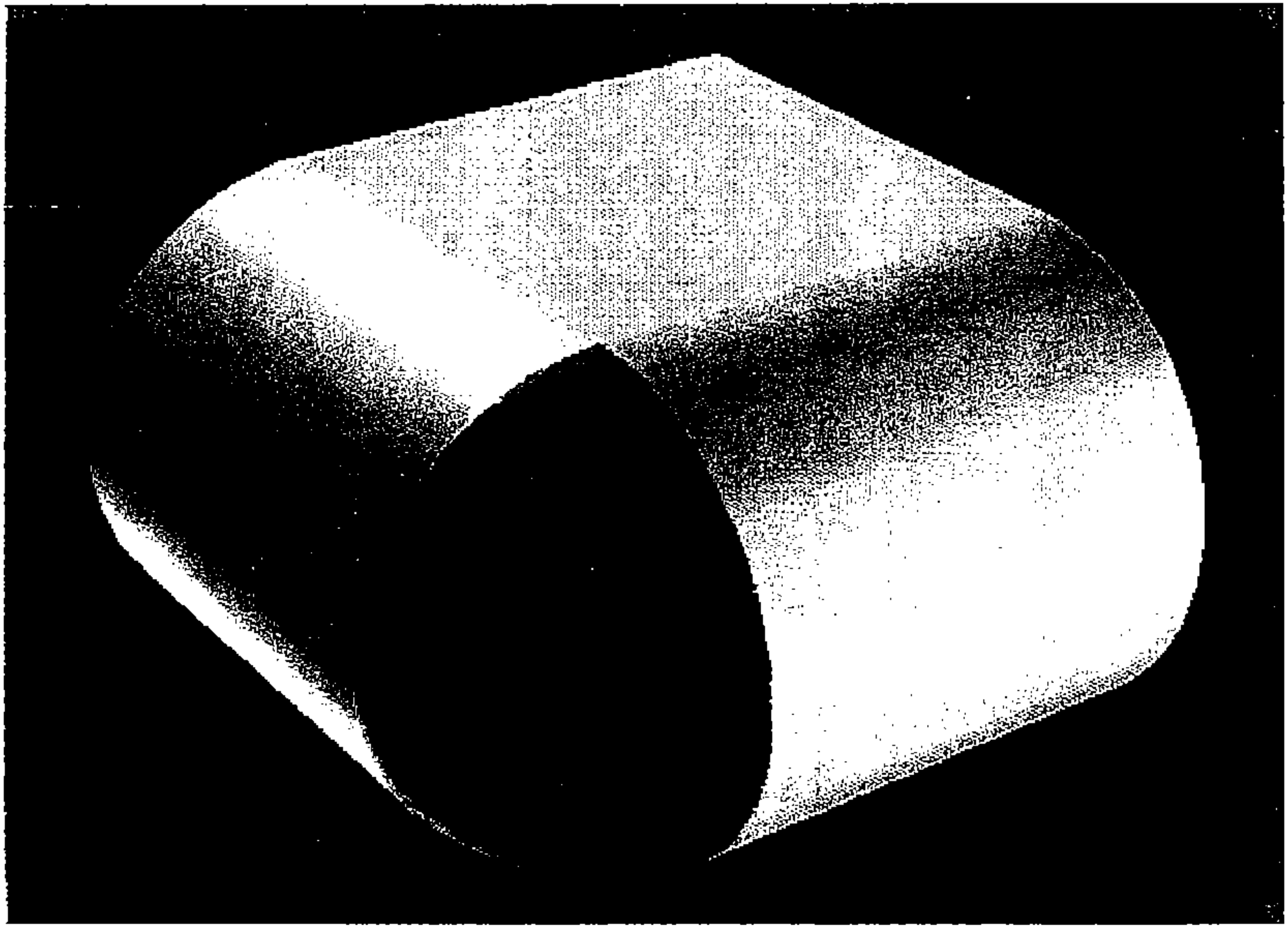


Figure 1: Corner Blending with a Quartic Surface

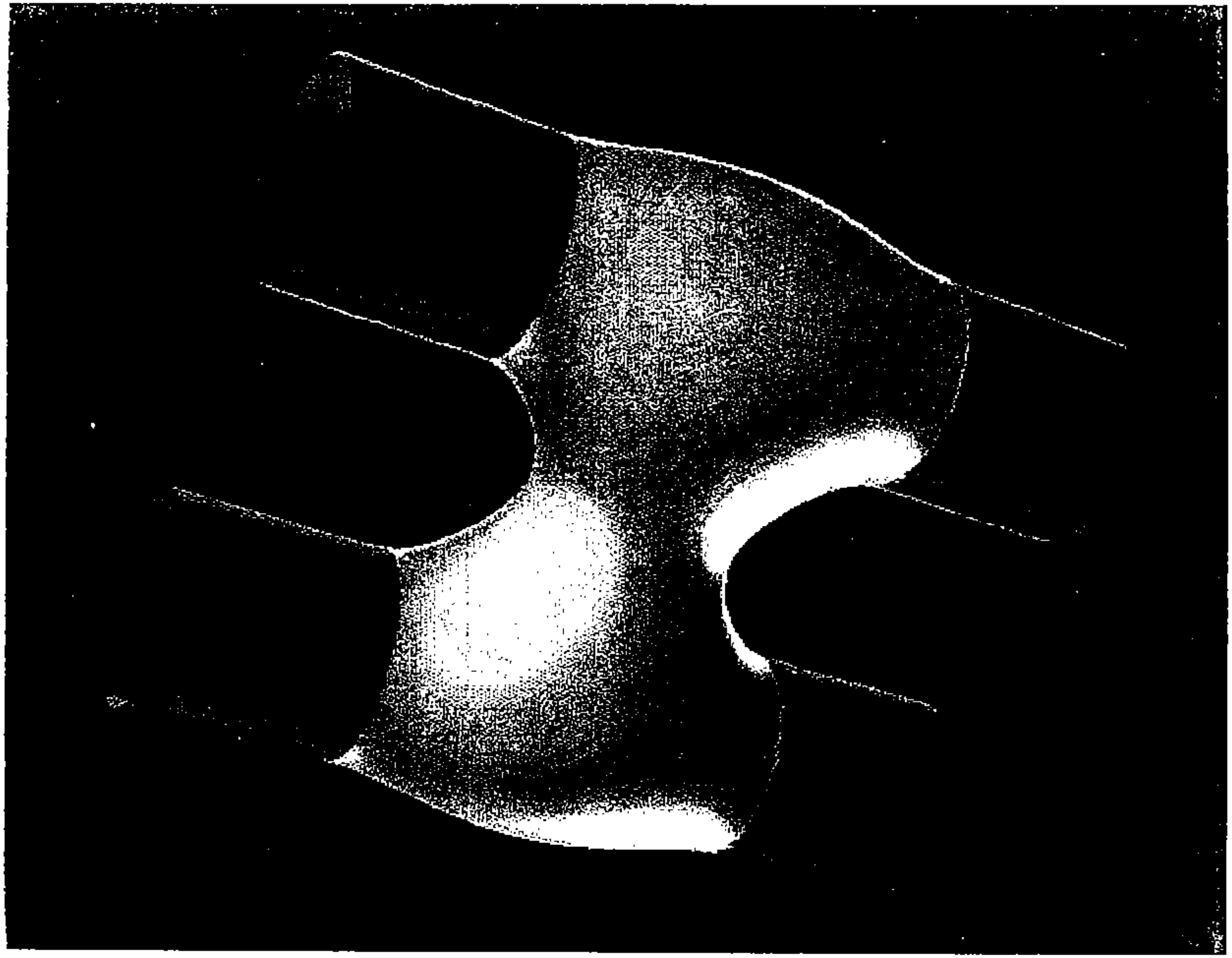


Figure 2: C^1-Join of Four Cylinders with a Quartic Surface

~/Jpic/sec.gif

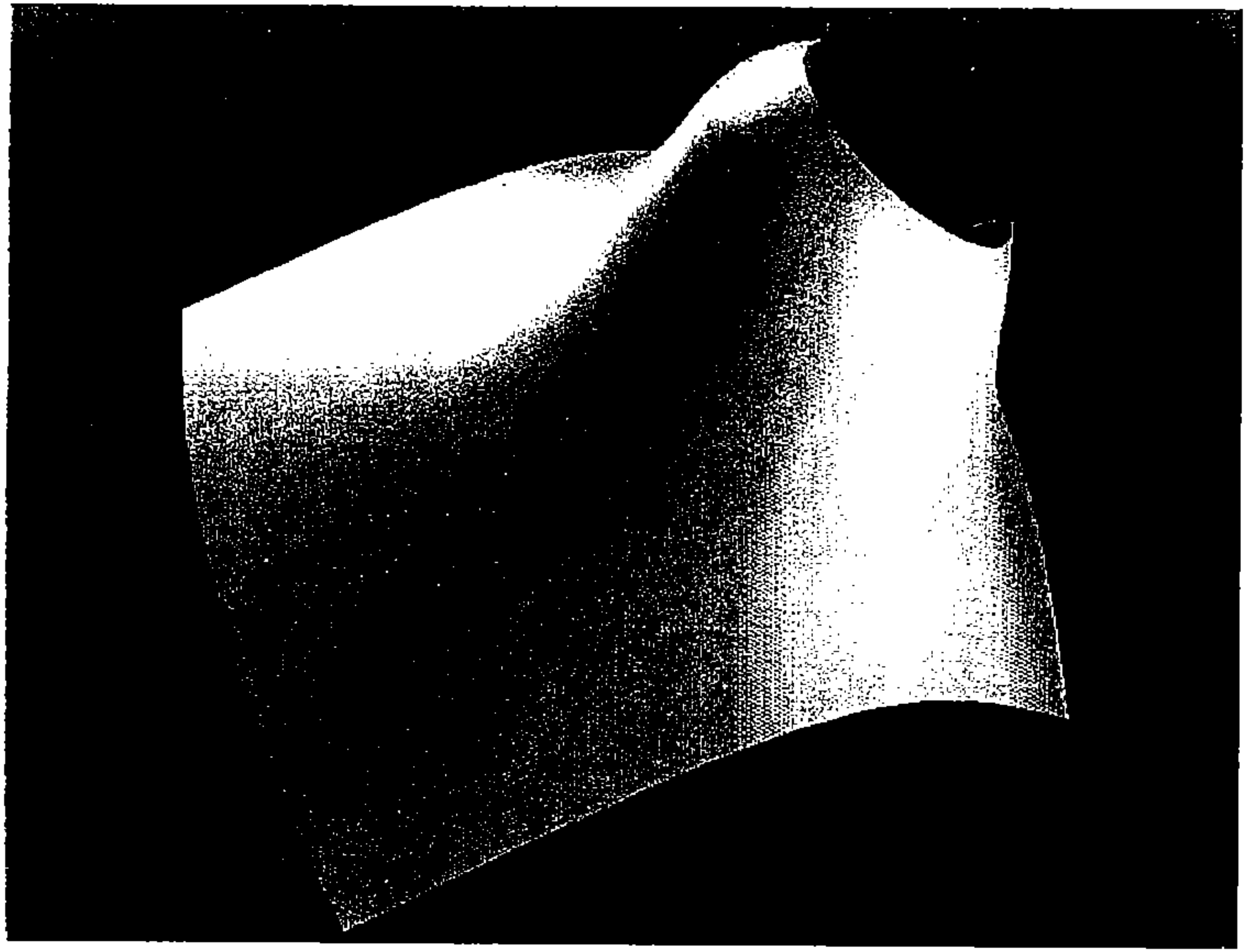


Figure 3: Example of a G^2-Continuous Surfaces

~tppi/gsces.
g-f

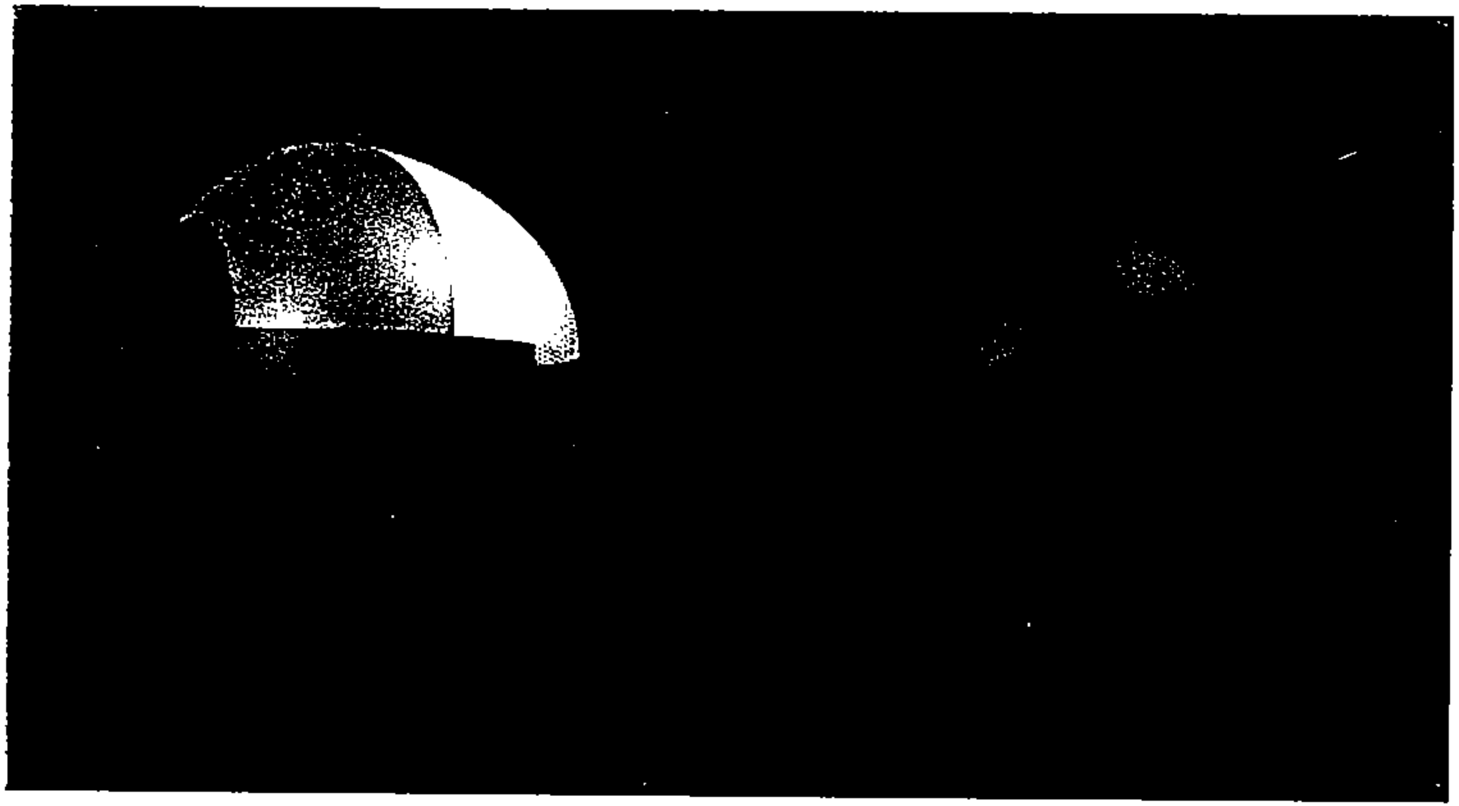
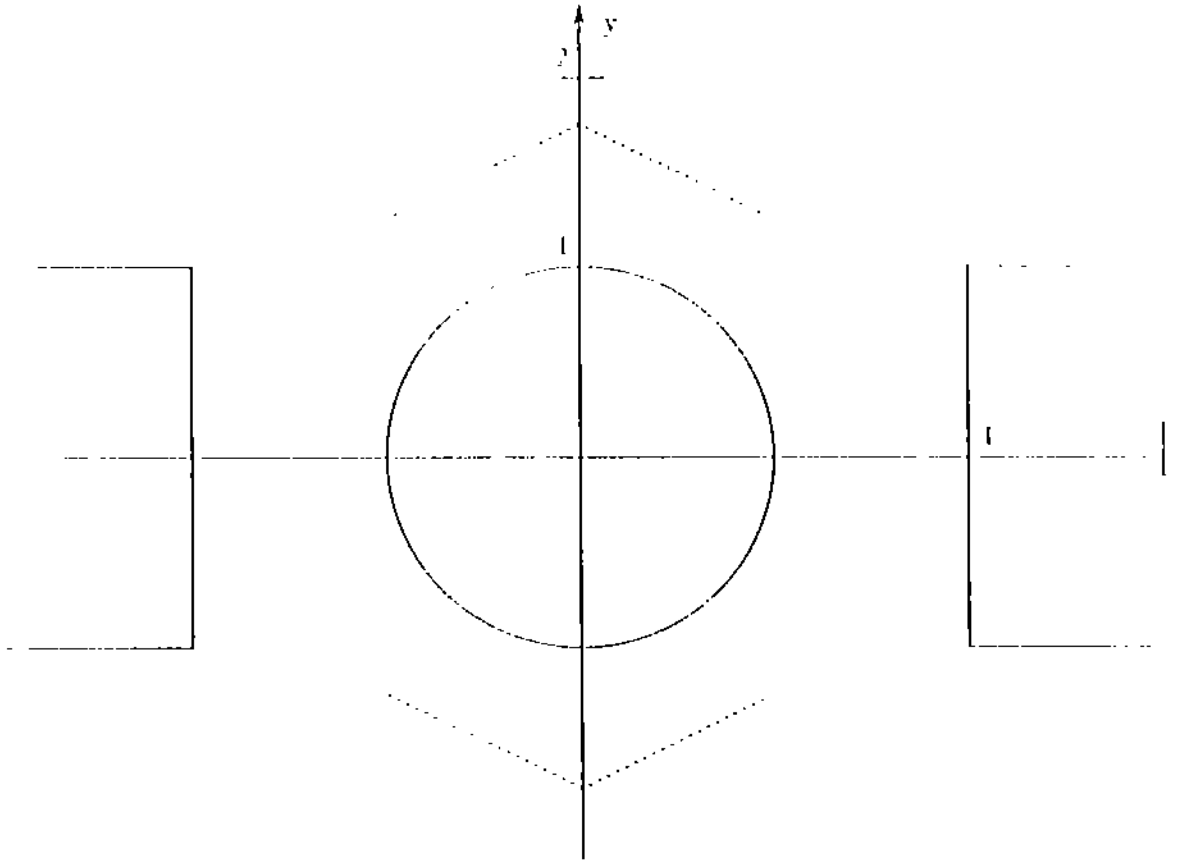
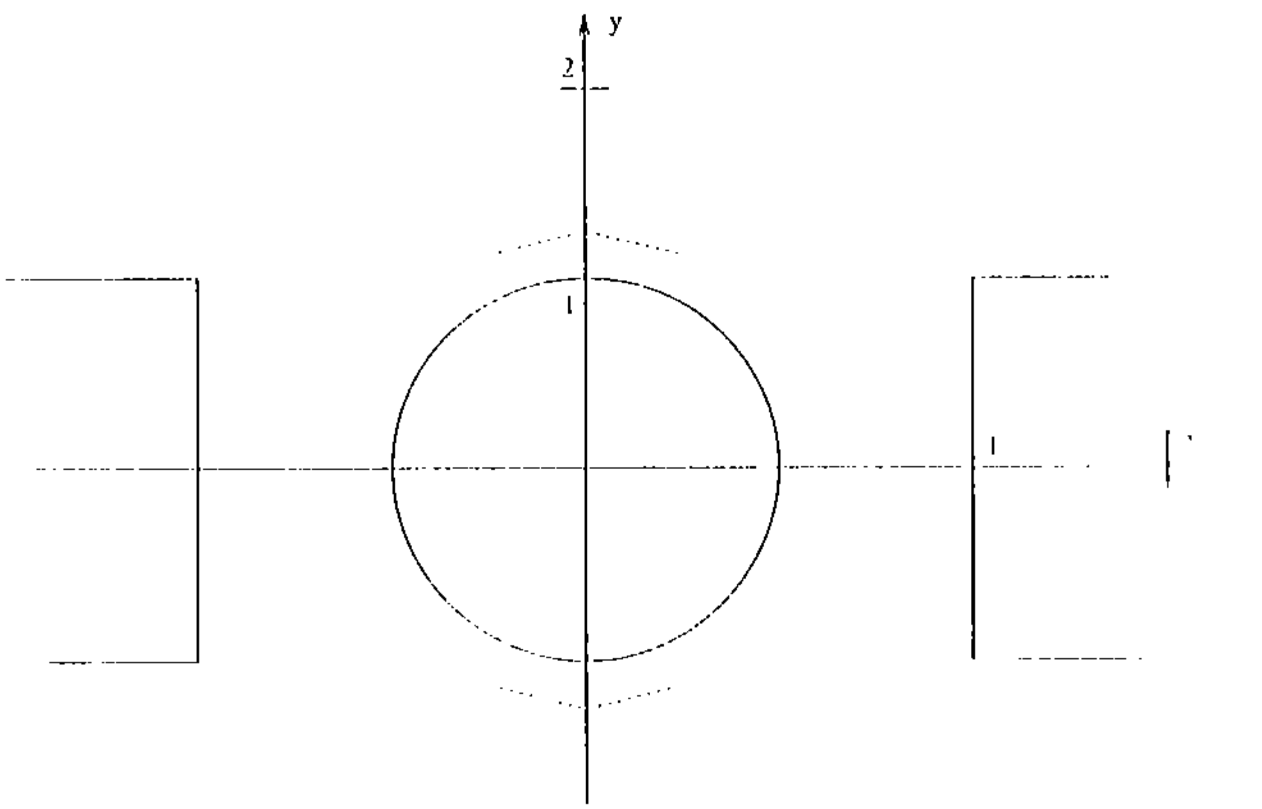


Figure 4: Example of a G^3-Continuous Surfaces



(a) Six Points in S_1



(b) Six Points in S_2

Figure 5: Points to be Approximated (a) (b)

Figure 6: C^1 interpolation with two different C^0 least-squares approximations (a) (b)

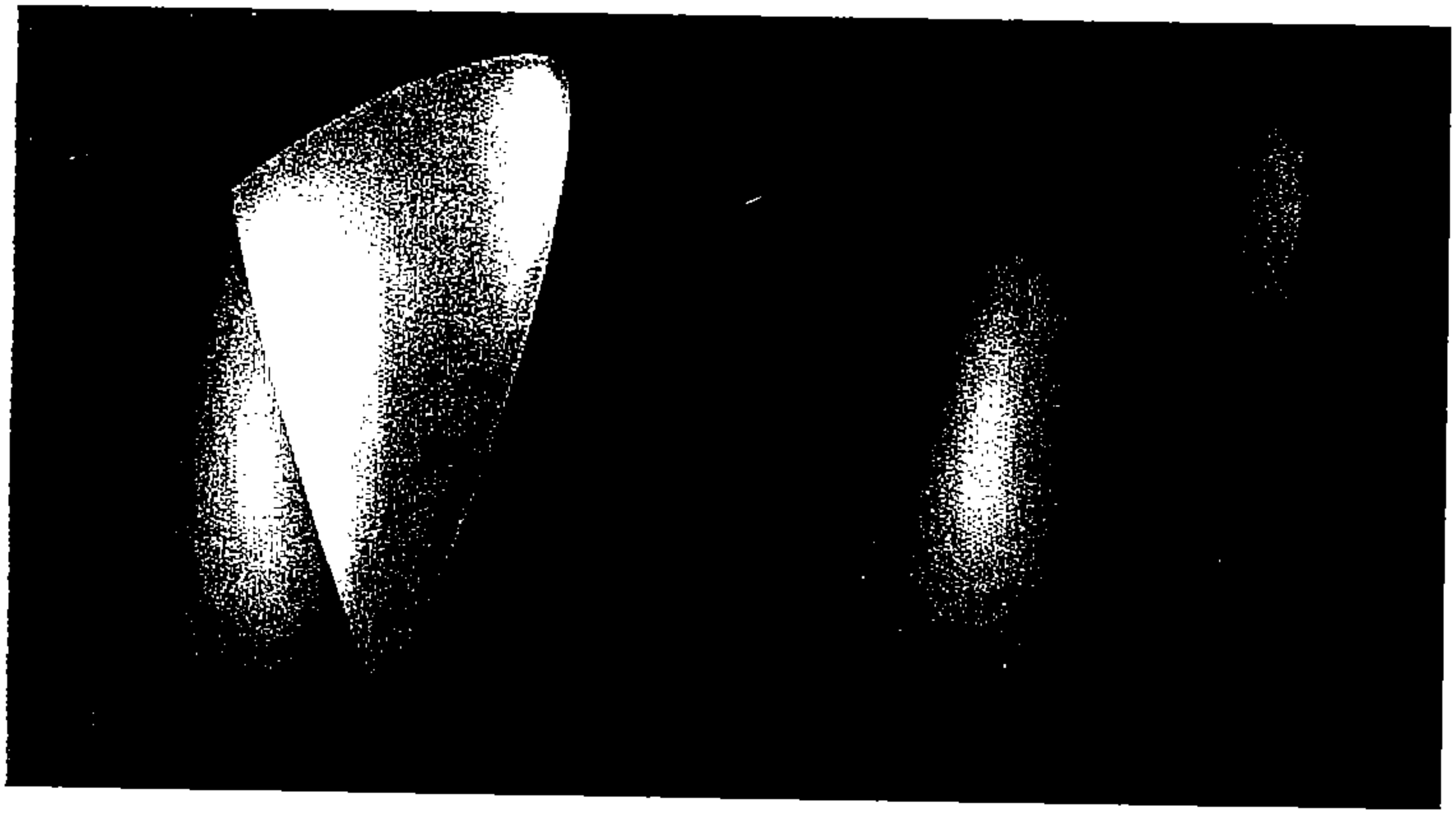


Figure 7: C^0 interpolation with C^1 least-squares approximation

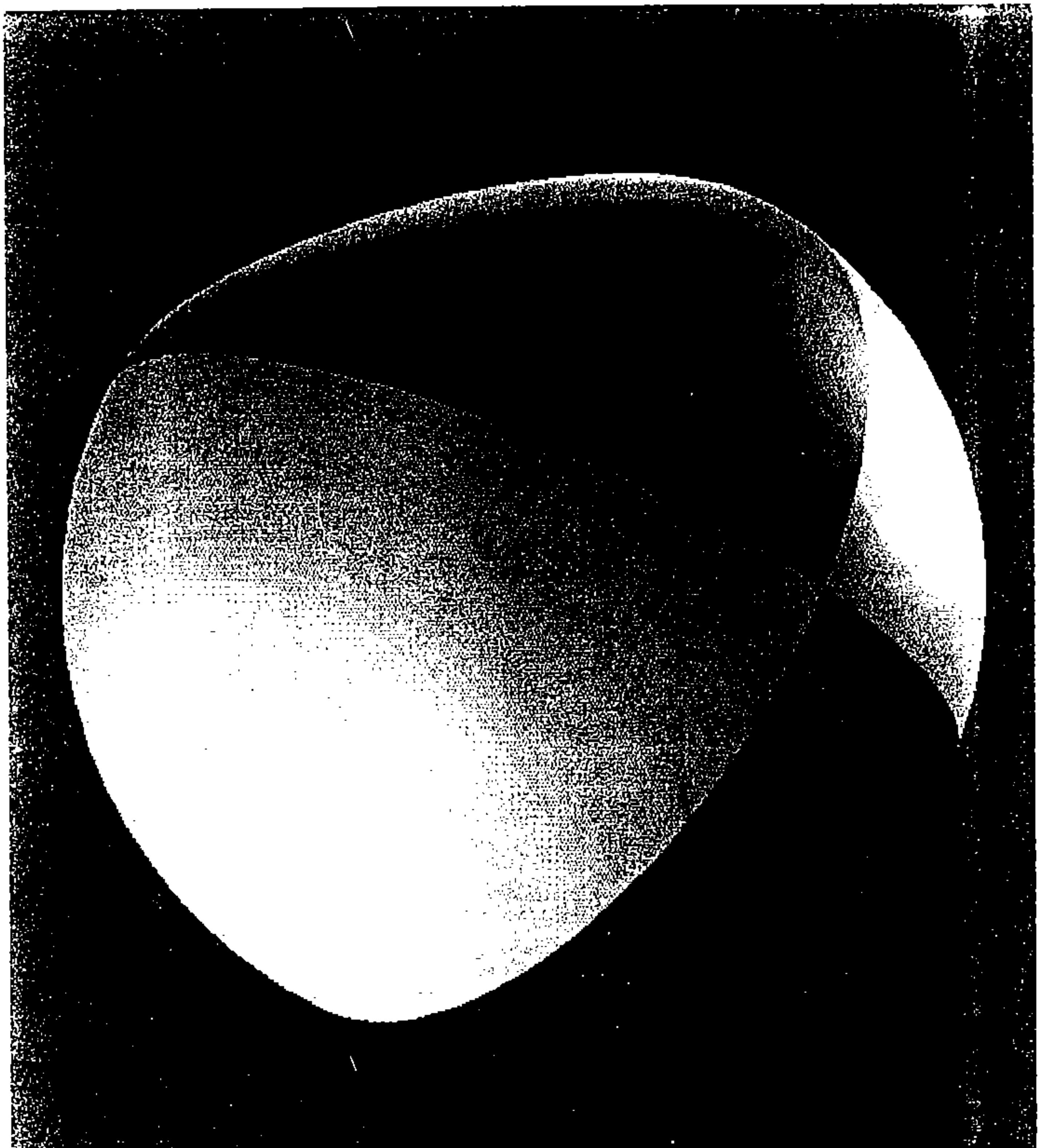


Figure 8: C^1 Mesh of Quadric and Quartic Patches

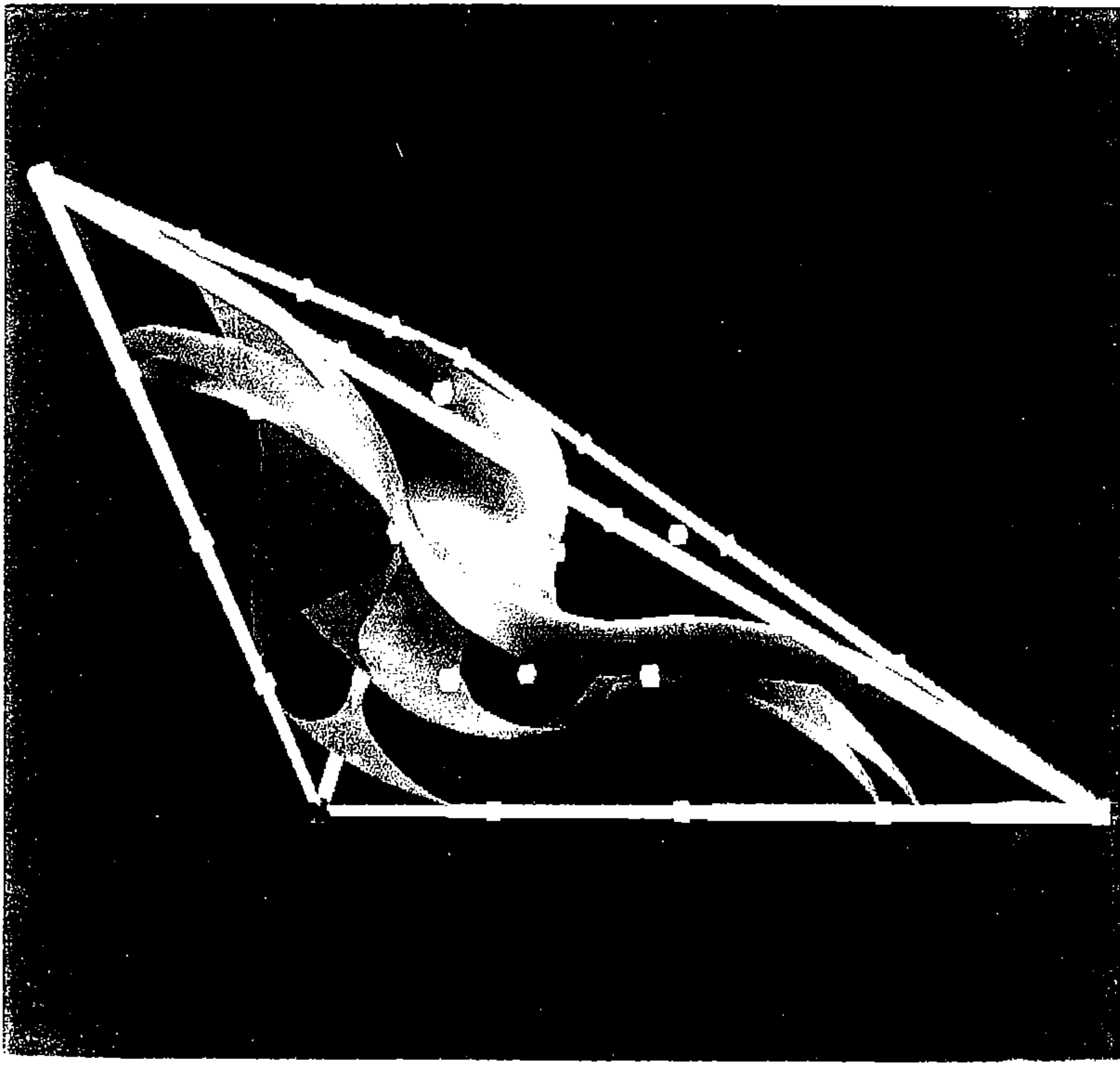


Figure 11: Interactive Shape Control of a Family of Quartic Surfaces

~/Jpic/
pops.grf

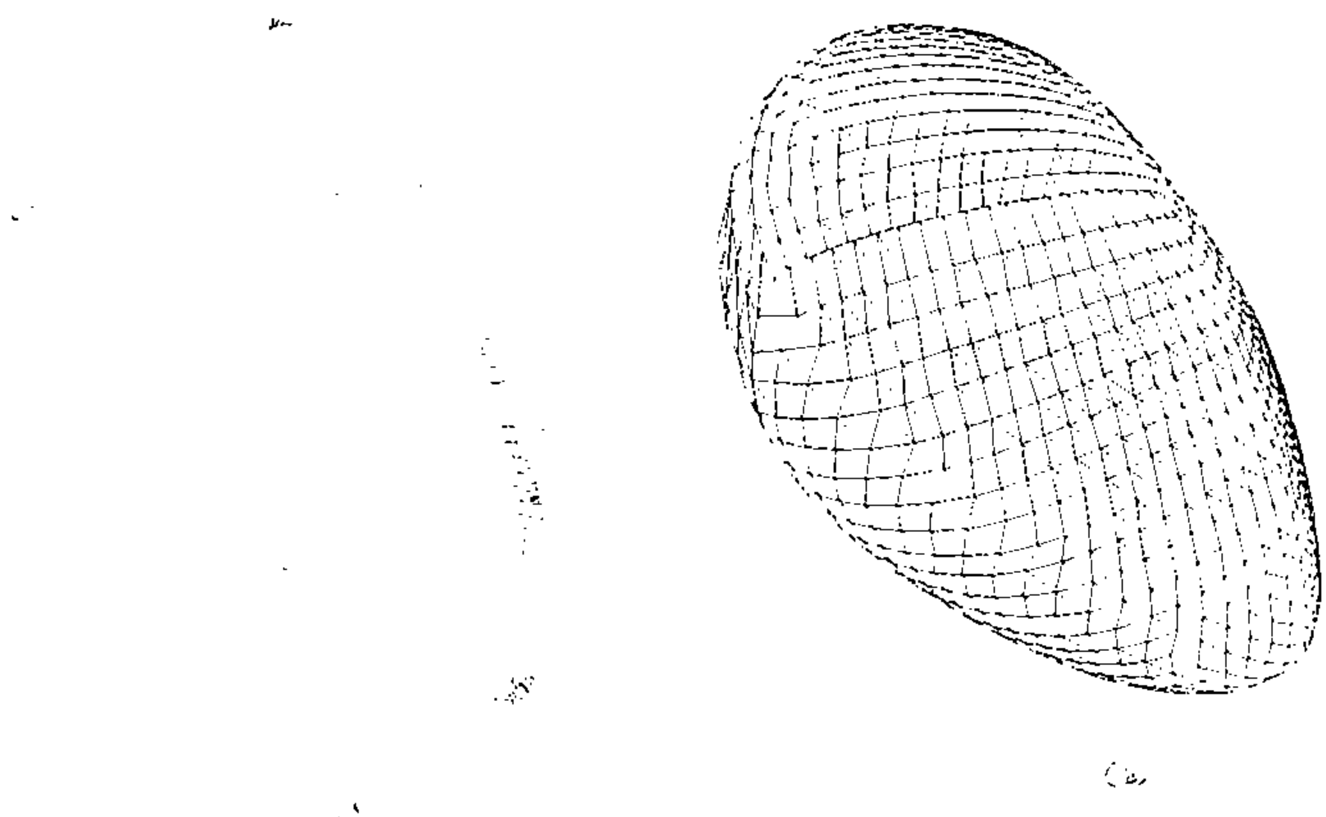


Figure 10: Locally supported triangular C^1 interpolants for smoothing polyhedra

Figure 9: C^1 Mesh of Quadric Patches