

1990

Algebraic Surface Design with Hermite Interpolation

Chandrajit L. Bajaj

Insung Ihm

Report Number:
90-939

Bajaj, Chandrajit L. and Ihm, Insung, "Algebraic Surface Design with Hermite Interpolation" (1990).
Department of Computer Science Technical Reports. Paper 798.
<https://docs.lib.purdue.edu/cstech/798>

This document has been made available through Purdue e-Pubs, a service of the Purdue University Libraries.
Please contact epubs@purdue.edu for additional information.

**ALGEBRAIC SURFACE DESIGN
WITH HERMITE INTERPOLATION**

**Chandrajit L. Bajaj
Insung Ihm**

**CSD TR-939
(Revised 12/90)**

Algebraic Surface Design with Hermite Interpolation

Chanderjit L. Bajaj*

Insung Ihm†

Department of Computer Sciences
Purdue University
West Lafayette, IN 47907

Abstract

This paper presents an efficient algorithm for constructing the lowest degree algebraic surfaces, which contain with C^1 continuity, any given collection of points and algebraic space curves having derivative information. Positional as well as derivative constraints on interpolating implicitly defined algebraic surfaces are translated into a *homogeneous linear system*, where the unknowns are the coefficients of the polynomial defining the algebraic surface. The containment of given points and curves with C^1 continuity, termed Hermite interpolation, yields tangent plane continuous surface fits. Computational details of the Hermite interpolation algorithm are also presented along with several illustrative applications of the interpolation technique to the construction of *joining* or *blending* surfaces for solid models as well as *fleshing* surfaces for curved wireframe models. A heuristic approach to interactive shape control of implicit algebraic surfaces is also given, and various open problems in algebraic surface design are discussed.

1 Introduction

While developing a solid modeling system for the construction of accurate computer models of solid physical objects [2], we have designed a technique for automatically generating low degree real algebraic surfaces, which yield a *tangent plane continuous* mesh of algebraic surface patches. Modeled physical objects with algebraic surface patches of low degrees, lend themselves to effective computations in several geometric modeling operations [4, 13, 16, 17, 18, 20, 21, 24, 29] as well as in tasks such as computer graphics display, animation, and physical simulations [8, 9, 15, 26].

In this paper, we present solutions to the problem of C^1 data interpolation using the lowest degree, implicitly represented algebraic surfaces in three dimensional real space \mathbb{R}^3 . An algebraic surface S in \mathbb{R}^3 is implicitly defined by a single polynomial equation $f(x, y, z) = 0$, where coefficients of f are over \mathbb{R} . The class of algebraic surfaces have the advantage of closure under several geometric operations (intersections, union, offset, etc.) often required in a solid modeling system. Furthermore, we choose the implicit equation representation for interpolation design since it captures all elements in the class of algebraic surfaces. This is contrasted with rational parametric

*Supported in part by NSF Grant DMS 88-16286, and ONR contract N00014-88-K-0402.

†Supported in part by a David Ross Fellowship.

equation representations wherein only a subset of algebraic surfaces can be defined by the trio of x , y and z given explicitly as rational functions of two parameters.

Prior approaches to C^1 interpolation and surface fitting, have focused primarily on the parametric representation of surfaces [7, 10, 12, 22, 30]. A good exposition of exact and least squares fitting of implicit algebraic surfaces through given data points, is presented in [19]. Sederberg [25] presents the idea of C^0 interpolation of data points and curves using implicitly defined algebraic surfaces. Here we extend the results of both these papers by using implicit algebraic surfaces for the C^1 interpolation of data points and space curves (defined implicitly as the common intersection of algebraic surfaces or in the rational parametric form) possibly having associated "normal" directions.

Specifically, the collection of data points are in \mathbb{R}^3 with or without associated "normal" directions. Moreover the data also may consist of a collection of algebraic space curves in \mathbb{R}^3 also with or without associated "normal" directions, varying along the entire span of the curves. Geometric properties for a surface to be designed are presented in terms of a collection of points and curves, with or without "normal" directions. Both points and space curves have an infinity of potential "normal" directions. While for points the directions may be chosen arbitrarily, for space curves, the varying "normal" directions are chosen to be orthogonal to the tangent vectors of the curves along the entire span. Our emphasis being algebraic space curves, the variance of the curves "normals" are restricted to polynomials of some degree. Also, we assume that any of the "normal" directions are never identically zero, a phenomenon that occurs at point and curve singularities. The C^1 interpolation algorithm, which we term Hermite interpolation, translates the C^1 interpolation constraints into a homogeneous linear system where the unknowns are coefficients of the polynomial defining the algebraic surface. The algorithm is thus a natural generalization into three dimensions of the conventional Hermite interpolation, applied to fitting curves through point data, and equating derivatives at those points. It should be, however, noted that our Hermite interpolation algorithm uses tangent plane information of points and curves, not their derivative information.

The rest of the paper is structured as follows. Section 2 presents some fundamental definitions and a key theorem used in the interpolation algorithm. In Section 3 and 4, the basic Hermite interpolation algorithm for implicit algebraic surfaces is presented. In Section 5 we prove that the Hermite interpolation determines all the algebraic surfaces which could C^1 interpolate the given set. An alternate formulation of tangent plane continuity, namely G^1 continuity, is also shown to be equivalent. Section 6 considers computational details of the algorithm implementation. Furthermore several example applications are presented to construct the lowest degree Hermite interpolating surfaces for "joining" and "blending" primary surfaces of solid models as well as for "fleshing" curved wire frame models of physical objects. Section 7 presents a heuristic method using a tetrahedral net with control weights to interactively and intuitively select and shape desirable instances from a family of C^1 interpolating algebraic surfaces. Finally, conclusions and open problems in algebraic surface design are discussed in Section 8.

2 Preliminaries

In this section, we give brief definitions of certain terms we need and also state a form of *Bezout's* theorem. For detailed and additional definitions, refer to [1, 27].

For any multivariate polynomial f , partial derivatives are written by subscripting, for example,

$f_x = \partial f / \partial x$, $f_{xy} = \partial^2 f / (\partial x \partial y)$, and so on. Since we consider algebraic curves and surfaces, we have $f_{xy} = f_{yx}$ etc. Vectors and vector functions are denoted by bold letters. An *algebraic surface* in \mathbb{R}^3 is implicitly defined by a single polynomial equation $f(x, y, z) = \sum c_{ijk} x^i y^j z^k = 0$, where the coefficients c_{ijk} of f are real numbers. The *normal* or *gradient* of $f(x, y, z) = 0$ is the vector $\nabla f = (f_x, f_y, f_z)$. A point $\mathbf{p} = (x_0, y_0, z_0)$ on a surface is a *regular point* if the gradient at \mathbf{p} is not null; otherwise the point is *singular*. An algebraic surface $f(x, y, z) = 0$ is *irreducible* if $f(x, y, z)$ does not factor over the field of complex numbers. An *algebraic space curve* is defined by the common intersection of two or more algebraic surfaces. Although a complete algebraic space curve in general cannot be completely determined by the intersection of only two surfaces, in geometric design we often restrict our consideration to a specific curve segment which is contained in the intersection of exactly two algebraic surfaces. A *rational parametric space curve* is represented by the triple $\mathbf{G}(s) = (x = G_1(s), y = G_2(s), z = G_3(s))$, where G_1, G_2 and G_3 are rational functions in s . We assume that the curve is 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 *degree* of an algebraic surface is the number of intersections between the surface and a line, counting complex, infinite and multiple intersections. This degree is also the same as the degree of the defining polynomial. The *degree* of an algebraic space curve is the number of intersections between the curve and a plane, counting complex, infinite and multiple intersections. The degree of an algebraic curve segment given as the intersection curve of two algebraic surfaces is also no larger than the product of the degrees of the two surfaces. Furthermore, the degree of a rational algebraic curve is the same as the maximum degree of the numerator and denominator polynomials in the defining triple of rational functions.

The followings are definitions pertinent to our Hermite interpolation algorithm :

Definition 2.1 Let $\mathbf{p} = (a, b, c)$ be a point with an associated "normal" $\mathbf{m} = (m_x, m_y, m_z)$ in \mathbb{R}^3 . An algebraic surface $S : f(x, y, z) = 0$ is said to contain \mathbf{p} with C^1 continuity if

(1) $f(\mathbf{p}) = f(a, b, c) = 0$, (containment condition)

and

(2) $\nabla f(\mathbf{p})$ is not zero and $\nabla f(\mathbf{p}) = \alpha \mathbf{m}$, for some nonzero α . (tangency condition)

Definition 2.2 Let C be an algebraic space curve with an associated varying "normal" $\mathbf{n}(x, y, z) = (n_x(x, y, z), n_y(x, y, z), n_z(x, y, z))$, defined for all points on C . An algebraic surface $S : f(x, y, z) = 0$ is said to contain C with C^1 continuity if

(1) $f(\mathbf{p}) = 0$ for all points \mathbf{p} of C . (containment condition)

and

(2) $\nabla f(\mathbf{p})$ is not identically zero and $\nabla f(\mathbf{p}) = \alpha \mathbf{n}(\mathbf{p})$, for some α and for all points \mathbf{p} of C . (tangency condition)

Definition 2.3 An algebraic surface $S : f(x, y, z) = 0$ is said to Hermite interpolate a given collection of data points with associated "normals", and data curves with associated "normals", if S contains all the data points and curves with C^1 continuity.

The following is one form of Bezout's theorem, the oldest theorem of algebraic geometry. As will be seen, this theorem plays a key role in proving the correctness of the Hermite interpolation algorithm.

Theorem 2.1 (Bezout) An algebraic curve C of degree d intersects an algebraic surface S of degree n in exactly nd points, counting complex, infinite, and multiple intersections, or C intersects S infinitely often, that is, a component of C lies entirely on S .

3 Interpolation of Points with Normals

3.1 Containment

From the containment condition of Definition 2.1, it follows that any algebraic surface $S : f(x, y, z) = 0$, whose coefficients satisfy the linear equation $f(\mathbf{p}) = 0$ will contain the point \mathbf{p} . For a set of k data points this yields k linear equations. An algebraic surface of degree n , has $K = \binom{n+3}{3} - 1$ independent coefficients. Interpolation of all the data points is achieved by selecting an algebraic surface of the smallest degree n such that $K \geq r$, where $r (\leq k)$ is the rank of a system of k linear equations. Similar approaches for constructing algebraic surfaces interpolating points can be found in [19].

3.2 Containment with Tangency

A point $\mathbf{p} = (a, b, c)$ with a "normal" $\mathbf{n} = (n_x, n_y, n_z)$ determines a unique plane $P : n_x x + n_y y + n_z z - (n_x a + n_y b + n_z c) = 0$, at the point \mathbf{p} . An algebraic surface $S : f(x, y, z) = 0$ of degree n that Hermite interpolates the point \mathbf{p} , can be constructed by setting up a linear system of equations as follows:

For each point \mathbf{p} with "normal" $\mathbf{n} = (n_x, n_y, n_z)$,

1. (**containment condition**) Use the linear equation $f(\mathbf{p}) = 0$ in the unknown coefficients of S .
2. (**tangency condition**) Select one of the following :
 - (a) If $n_x \neq 0$, use the equations $n_x f_y(\mathbf{p}) - n_y f_x(\mathbf{p}) = 0$ and $n_x f_z(\mathbf{p}) - n_z f_x(\mathbf{p}) = 0$.
 - (b) If $n_y \neq 0$, use the equations $n_y f_x(\mathbf{p}) - n_x f_y(\mathbf{p}) = 0$ and $n_y f_z(\mathbf{p}) - n_z f_y(\mathbf{p}) = 0$.
 - (c) If $n_z \neq 0$, use the equations $n_z f_x(\mathbf{p}) - n_x f_z(\mathbf{p}) = 0$ and $n_y f_z(\mathbf{p}) - n_z f_y(\mathbf{p}) = 0$.
3. Next ensure that the coefficients of $f(x, y, z) = 0$ satisfying the above three linear equations, additionally satisfy the linear constraints $\nabla f(\mathbf{p}) \neq 0$, since non-tangency at \mathbf{p} may occur if S turns out to be singular at \mathbf{p} .

The proof of correctness of the above algorithm follows from the following lemma.

Lemma 3.1 *The equations of the above algorithm satisfy Definition 2.1 of point containment and tangency.*

Proof : The first linear equation $f(\mathbf{p}) = 0$ satisfies containment by definition. We now show that the remaining equations satisfy $\nabla f(\mathbf{p}) = \alpha \cdot \mathbf{n}$ for a nonzero α . Since \mathbf{n} is not the $(0, 0, 0)$ vector, without loss of generality, we may assume that $n_x \neq 0$ in step 2. above. Other cases of $n_y \neq 0$ or $n_z \neq 0$ can be handled analogously. Now let $\alpha = \frac{f_x}{n_x}$, assuming $n_x \neq 0$. Then $f_x = \alpha \cdot n_x$ and substituting it in the selected linear equation $n_x f_y - n_y f_x = 0$ yields $f_y = \alpha \cdot n_y$ and substituting it again in the other selected linear equation $n_x f_z - n_z f_x = 0$ yields $f_z = \alpha \cdot n_z$. Hence $\nabla f(\mathbf{p}) = \alpha \cdot \mathbf{n}$. Finally, note that $f_x = 0$ for $n_x \neq 0$, in the selected linear equations of step 2 (a), would cause $\nabla f(\mathbf{p}) = 0$, which we ensured would not happen in step 3 of the algorithm. Hence $f_x \neq 0$ and so $\alpha \neq 0$ and the lemma is proved. \square

4 Interpolation of Curves with Normals

The varying “normal” associated with a space curve C can be defined implicitly by the triple $\mathbf{n}(x, y, z) = (n_x(x, y, z), n_y(x, y, z), n_z(x, y, z))$ where n_x , n_y and n_z are polynomials of maximum degree m and defined for all points $\mathbf{p} = (x, y, z)$ along the curve C . For the special case of a rational curve which we shall treat separately in Subsections 4.1.2 and 4.2.2, the varying “normals” can also be defined parametrically as $\mathbf{n}(s) = (x = n_x(s), y = n_y(s), z = n_z(s))$, with n_x , n_y and n_z now rational functions in s .

4.1 Containment

4.1.1 Algebraic Curves: Implicit Definition

Let $C : (f_1(x, y, z) = 0, f_2(x, y, z) = 0)$ implicitly define an algebraic space curve of degree d . The irreducibility of the curve is not a restriction, since reducible curves can be handled by treating each irreducible curve component separately. For precise definitions of irreducible components of an algebraic curve, see, for example [27]. The containment condition (as well as the tangency condition) requires the interpolating surface to be zero at a finite number of points on the curve. To ensure containment of a specific irreducible component then requires choosing this finite number of points on that component. The precise number, derived from Bezout’s theorem, is a linear function of the *degree* of that curve component.

The situation is more complicated in the real setting, if we wish to achieve separate containment of one of possibly several connected real ovals of a single irreducible component of the space curve. There is first of course the nontrivial problem of specifying a single isolated real oval of a curve. See [3] where a solution is derived in terms of a decomposition of space into cylindrical cells which separate out the various components of any real curve (or any real algebraic or semi-algebraic set).

An interpolating surface $S : f(x, y, z) = 0$ of degree n for containment of an irreducible curve component C , is computed as follows:

1. Choose a set L_c of $nd + 1$ points on C , $L_c = \{\mathbf{p}_i = (x_i, y_i, z_i) | i = 1, \dots, nd + 1\}$. The set L_c may be computed, for example, by tracing the intersection of $f_1 = f_2 = 0$ ([5]). Thus, alternatively, an algebraic curve may be given as a list of points.
2. Next, set up $nd + 1$ homogeneous linear equations $f(\mathbf{p}_i) = 0$, for $\mathbf{p}_i \in L_c$. Any nontrivial solution of this linear system will represent an algebraic surface which interpolates the entire curve C .

The proof of correctness of the above algorithm is captured in the following Lemma.

Lemma 4.1 *To satisfy the containment condition of an algebraic curve C of degree d by an algebraic surface S of degree n , it suffices to satisfy the containment condition of $nd + 1$ points of C by S .*

Proof: This is essentially a restatement of Bezout’s theorem of Section 2. By making S contain $nd + 1$ points of C , ensures that S must intersect C infinitely often and since C is irreducible, S must contain the entire curve. \square

Remember $S : f(x, y, z) = 0$ of degree n has $K = \binom{n+3}{3} - 1$ independent coefficient unknowns. Let r be the rank of the system of $nd + 1$ linear equations. There are non-trivial solutions to this

homogeneous system if and only if $K > r$ and a unique non-trivial solution when $K = r$. Hence, again an interpolating surface can be obtained by choosing the smallest n such that $K \geq r$.

4.1.2 Rational Curves : Parametric Definition

When a curve is given in rational parametric form, its equations can be used directly to produce a linear system for interpolation, instead of first computing $nd + 1$ points on the curve. Let $C : (x = G_1(t), y = G_2(t), z = G_3(t))$ be a rational curve of degree d . An interpolating surface $S : f(x, y, z) = 0$ of degree n which contains C is computed as follows:

1. Substitute $(x = G_1(t), y = G_2(t), z = G_3(t))$ into the equation $f(x, y, z) = 0$.
2. Simplify and rationalize the expression from step 1. to obtain the numerator $Q(t) = 0$, where Q is a polynomial in t , of degree at most nd , and with coefficients which are linear expressions in the coefficients of f . For Q to be identically zero, each of its coefficients must be zero, and hence we obtain a system of at most $nd + 1$ linear equations, where the unknowns are the coefficients of f . Any non-trivial solution of this linear system will represent a surface S which interpolates C .

Lemma 4.2 *The containment condition is satisfied by step 2 of the above algorithm.*

Proof Obvious. \square

4.2 Containment with Tangency

In order to Hermite interpolate an algebraic curve C with associated "normals" \mathbf{n} by an algebraic surface S , we need to again solve a homogeneous linear system, whose equations stem from both the *containment condition* and the *tangency conditions* of Definition 2.2.

4.2.1 Algebraic Curves with Normals: Implicit Definition

As before, let $C : (f_1(x, y, z) = 0, f_2(x, y, z) = 0)$ implicitly define an irreducible algebraic space curve of degree d , together with associated "normals" defined implicitly by the triple $\mathbf{n}(x, y, z) = (n_x(x, y, z), n_y(x, y, z), n_z(x, y, z))$ where n_x, n_y and n_z are polynomials of maximum degree m and defined for all points $\mathbf{p} = (x, y, z)$ along the curve C . A Hermite interpolating surface $S : f(x, y, z) = 0$ of degree n which contains C with C^1 continuity is then computed as follows:

1. Choose a set L_c of $(n + m - 1)d + 1$ points on C , $L_c = \{\mathbf{p}_i = (x_i, y_i, z_i) | i = 1, \dots, (n + m - 1)d + 1\}$. The set L_c may be computed, as before, by tracing the intersection of $f_1 = f_2 = 0$ ([5]).
2. Construct the list L_t of $(n+m-1)d+1$ point-normal pairs on C , $L_t = \{(x_i, y_i, z_i), (n_{xi}, n_{yi}, n_{zi}) | i = 1, \dots, (n + m - 1)d + 1\}$, where $(n_{xi}, n_{yi}, n_{zi}) = \mathbf{n}(\mathbf{p}_i)$ for $\mathbf{p}_i \in L_c$. Thus, alternatively, an algebraic curve C and its associated "normals" \mathbf{n} may (either or both) be given as a list of points or point-normal pairs.
3. (**containment condition**) Next, set up $nd + 1$ homogeneous linear equations $f(\mathbf{p}_i) = 0$, for $\mathbf{p}_i \in L_c$, $i = 1, \dots, nd + 1$.

4. (tangency condition)

- (a) Compute $\mathbf{t}(x, y, z) = \nabla f_1(x, y, z) \times \nabla f_2(x, y, z)$. Note $\mathbf{t} = (t_x, t_y, t_z)$ is the tangent vector to C .
- (b) Select one of the following :
- i. If $t_x \neq 0$, use the equation $f_y \cdot n_z - n_y \cdot f_z = 0$.
 - ii. If $t_y \neq 0$, use the equation $f_x \cdot n_z - n_x \cdot f_z = 0$.
 - iii. If $t_z \neq 0$, use the equation $f_x \cdot n_y - n_x \cdot f_y = 0$.

Substitute each point-normal pair in L_t into the above selected equation to yield additionally $(n + m - 1)d + 1$ linear equations in the coefficients of the $f(x, y, z)$.

5. In total, we obtain a homogeneous system of $(2n+m-1)d+2$ linear equations. Any non-trivial solution of the linear system, for which additionally ∇f is not identically zero for all points of C , (that is, the surface S is not singular at all points along the curve C), will represent a surface which Hermite interpolates C .

The proof of correctness of the above algorithm follows from Lemma 4.1 and the following lemma, which shows why the selected equation of step 4(b) evaluated at $(n + m - 1)d + 1$ point-normal pairs, are sufficient.

Lemma 4.3 *To satisfy the tangency condition of an algebraic curve C of degree d with "normal" \mathbf{n} of degree m , by an algebraic surface S of degree n , it suffices to satisfy the tangency condition at $(n + m - 1)d + 1$ points of C by S as in step 4 of the above algorithm.*

Proof : In step 4(b) above, assume without loss of generality that $t_x \neq 0$. Then the selected equation

$$f_y \cdot n_z - n_y \cdot f_z = 0 \tag{1}$$

We first show that even though equation (1) is evaluated at only $(n + m - 1)d + 1$ points of C in step 4(b) above, it holds for all points on C . Equation (1) defines an algebraic surface T of degree $(n + m - 1)$ which intersects C of degree d at, at most, $(n + m - 1)d$ real points. Invoking Bezout's theorem, and from the irreducibility of C , it follows that C must lie entirely on the surface T . Hence equation (1) is valid along the entire curve C .

We now show that step 4 of the above algorithm, satisfies the tangency condition as specified in definition 2.2. Since \mathbf{t} of step 4(a) is a tangent vector at all points of C , and the surface $S : f = 0$ contains C , the gradient vector ∇f is orthogonal to \mathbf{t} , which yields the equation :

$$f_x \cdot t_x + f_y \cdot t_y + f_z \cdot t_z = 0 \tag{2}$$

valid for all points of C . Next, from the definition of a "normal" of a space curve,

$$n_x \cdot t_x + n_y \cdot t_y + n_z \cdot t_z = 0 \tag{3}$$

valid for all points of C . Now it is impossible that both $n_y(x, y, z)$ and $n_z(x, y, z)$ are identically zero along C , since if they were then equation (3) would imply that $n_x \cdot t_x = 0$, and as we had assumed that $t_x \neq 0$, would in turn imply that also $n_x = 0$ along C , which would contradict the

earlier assumption that \mathbf{n} is not identically zero. Hence, at least, one of n_y and n_z must also be nonzero. Without loss of generality, let $n_y \neq 0$. Also, let $\alpha(x, y, z) = \frac{f_y}{n_y}$. Then,

$$f_y = \alpha \cdot n_y \quad (4)$$

and substituting into equation (1) yields

$$f_x = \alpha \cdot n_x \quad (5)$$

for all points on C . From equations (2), (4) and (5) we obtain,

$$f_x \cdot t_x + \alpha \cdot n_y \cdot t_y + \alpha \cdot n_z \cdot t_z = 0 \quad (6)$$

By multiplying α to equation (3) and subtracting equation (6) from it, we obtain

$$f_x \cdot t_x = \alpha \cdot n_x \cdot t_x \quad (7)$$

and since $t_x \neq 0$, finally obtain

$$f_x = \alpha \cdot n_x \quad (8)$$

valid at all points of C . Hence equations (4), (5), and (8) together imply that $\nabla f(x, y, z) = \alpha \cdot \mathbf{n}$ for all points C and some nonzero α .¹ Hence, the tangency condition of definition 2.2 is met. \square

4.2.2 Rational Curves with Normals : Parametric Definition

When both a space curve and its associated "normal" are given in rational parametric form, their equations can be used directly to produce a linear system for interpolation, instead of first computing $(n + m - 1)d + 1$ points on the curve. Let $C : (x = G_1(s), y = G_2(s), z = G_3(s))$ be a rational curve of degree d with associated "normals" $\mathbf{n}(s) = (n_x(s), n_y(s), n_z(s))$ of degree m . A Hermite interpolating surface $S : f(x, y, z) = 0$ of degree n which contains C with C^1 continuity is computed as follows:

1. (**containment condition**) Substitute $(x = G_1(s), y = G_2(s), z = G_3(s))$ into the equation $f(x, y, z) = 0$. This results in, at most, $nd + 1$ homogeneous linear equations as in Subsection 4.1.2.
2. (**tangency condition**)
 - (a) Compute $\nabla f(s) = \nabla f(G_1(s), G_2(s), G_3(s))$ and $\mathbf{t}(s) = (\frac{dx}{ds}, \frac{dy}{ds}, \frac{dz}{ds})$. Note $\mathbf{t} = (t_x, t_y, t_z)$ is the tangent vector to C .
 - (b) Select one of the following:
 - i. If $t_x \neq 0$, use the equation $f_y(s) \cdot n_z(s) - n_y(s) \cdot f_z(s) = 0$.
 - ii. If $t_y \neq 0$, use the equation $f_x(s) \cdot n_z(s) - n_x(s) \cdot f_z(s) = 0$.
 - iii. If $t_z \neq 0$, use the equation $f_x(s) \cdot n_y(s) - n_x(s) \cdot f_y(s) = 0$.

¹From the equation (6) we see that $\alpha(x, y, z)$ must not be identically zero along C , for otherwise, $\nabla f = (0, 0, 0)$ for points along C and would contradict the fact that we chose a non-trivial solution for the surface $S : f = 0$ where ∇f is not identically zero.

In each case, the numerator of the simplified rational function equation is set to zero. This yields, additionally, at most, $(n - 1)d + m + 1$ linear equations in the coefficients of the surface $S : f(x, y, z) = 0$.

3. In total, we obtain a homogeneous system of, at most, $(2n - 1)d + m + 2$ linear equations. Any non-trivial solution of the linear system, for which additionally ∇f is not identically zero for all points of C , (that is, the surface S is not singular along the curve C), will represent a surface which Hermite interpolates C .

The proof of correctness of the above algorithm follows from Lemma 4.2 and the following lemma.

Lemma 4.4 *If we choose a nontrivial solution for which the resulting Hermite interpolating surface S is nonsingular along the given curve C , then the step 2 guarantees that the tangency condition of Definition 2.2 is met.*

Proof : The proof is very similar to that of Lemma 4.3 with minor modifications and is omitted. \square

5 Geometric Continuity

In the Hermite interpolation algorithm, C^1 continuity between two surfaces is achieved by making the tangent planes of the two surfaces identical at a point or at all points along the common curve of intersection. This definition of continuity agrees with several other definitions of geometric or G^1 continuity given for parametric and implicit algebraic surfaces. DeRose [11] gives a definition of higher orders of geometric continuity between parametric surfaces where two surfaces F_1 and F_2 meet with order k geometric continuity (shortly, G^k continuity) along a curve C if and only if there exist 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 [28] formulates an intuitive definition of G^k continuity between implicit surfaces as following :

Definition 5.1 *Two algebraic surfaces $f(x, y, z) = 0$ and $g(x, y, z) = 0$ meet with G^k rescaling continuity at a point p or along an algebraic curve C if and only if there exists two functions $a(x, y, z)$ 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 [28], 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 this are thus corrected by allowing multiplication by (rescaling) 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.

Obviously, the definition for G^0 rescaling continuity corresponds to the containment definition in Section 2. The following lemma shows that the C^1 continuity definition in Section 2 agrees with the G^1 rescaling continuity definition.

Lemma 5.1 G^1 rescaling continuity between $f(x, y, z) = 0$ and $g(x, y, z) = 0$ at a common point p or along a common curve C corresponds to $f(x, y, z) = 0$ and $g(x, y, z) = 0$ having common tangent planes at p or along every point of C .

Proof: The requirement for G^1 continuity is that there exist $a(x, y, z)$ and $b(x, y, z)$, not identically zero at p or along C , such that

$$\begin{aligned} \frac{\partial(af - bg)}{\partial x} &= a_x f + a f_x - b_x g - b g_x \\ &= 0 \quad \text{at } p \text{ or along } C, \\ \frac{\partial(af - bg)}{\partial y} &= a_y f + a f_y - b_y g - b g_y \\ &= 0 \quad \text{at } p \text{ or along } C, \\ \frac{\partial(af - bg)}{\partial z} &= a_z f + a f_z - b_z g - b g_z \\ &= 0 \quad \text{at } p \text{ or along } C. \end{aligned}$$

Since p or C is contained in both f and g (that is, $f = g = 0$ at p or along C), the requirement becomes

$$\begin{aligned} a f_x &= b g_x \\ a f_y &= b g_y \\ a f_z &= b g_z. \end{aligned}$$

which means $(f_x, f_y, f_z) = \frac{b}{a}(g_x, g_y, g_z)$ at p or along C . Hence, f and g are required to have common tangent planes at p or along C . \square

The correctness proofs in Section 3 and Section 4, implies that Hermite interpolation finds all the surfaces which have common tangent planes at a point or a curve. It also yields the following theorem.

Theorem 5.1 Hermite interpolation finds all the surfaces F which meet a surface G along a specified curve C on G with G^1 rescaling continuity.

Families of surfaces F as in the above theorem can be constructed in the Hermite interpolation framework of Section 4 as follows. Given a surface G and a curve C on G (defined implicitly or parametrically) the input to the Hermite interpolation algorithm is the curve C and the "normals" to the curve C obtained directly from the ∇G evaluated along C . The algorithm then yields a solution representation for the coefficients of the implicit algebraic surface families F which meet surface G along C with C^1 or G^1 rescaling continuity. Several examples of this are provided in the next section.

6 Applications: Mixed Point and Space Curve Data

The basic mechanics of Hermite interpolation using algebraic surfaces, as presented in the algorithms of Section 3 and Section 4, are

1. properties of a surface to be designed are described in terms of a combination of points, curves, and possibly associated "normal" directions,
2. these properties are translated into a homogeneous linear system of equations with extra surface constraints, and then
3. nontrivial solutions of the above system are computed.

In this section, we discuss some computational aspects of Hermite interpolation, and give several examples of algebraic surface design with Hermite interpolation.

6.1 On Computing Nontrivial Solutions

As explained in the previous sections, Hermite interpolation algorithm converts geometric properties of a surface into a homogeneous linear system :

$$Ax = 0 \quad (A \in \mathbb{R}^{p \times q}, \mathbf{x} \in \mathbb{R}^q),$$

where p is the number of equations generated, q is the number of unknown coefficients of a surface of degree n (hence, $q = \binom{n+3}{3}$), A is a matrix for linear equations, and \mathbf{x} is a vector whose elements are unknown coefficients.

In order to solve the linear system in a computationally stable manner, we compute the singular value decomposition (SVD) of A [14]. Hence, A is decomposed as $A = U\Sigma V^T$ where $U \in \mathbb{R}^{p \times p}$ and $V \in \mathbb{R}^{q \times q}$ are orthogonal matrices, and $\Sigma = \text{diag}(\sigma_1, \sigma_2, \dots, \sigma_r) \in \mathbb{R}^{p \times q}$ is a diagonal matrix with diagonal elements $\sigma_1 \geq \sigma_2 \geq \dots \geq \sigma_r \geq 0$ ($r = \min\{p, q\}$). It can be proved that the rank s of A is the number of the positive diagonal elements of Σ , and that the last $q - s$ columns of V span the null space of A . Hence, the nontrivial solutions of the homogeneous linear system are compactly expressed as :

$$\{\mathbf{x} (\neq \mathbf{0}) \in \mathbb{R}^q \mid \mathbf{x} = \sum_{i=1}^{q-s} \tau_i \cdot \mathbf{v}_{s+i}, \text{ where } \tau_i \in \mathbb{R}, \text{ and } \mathbf{v}_j \text{ is the } j\text{th column of } V\}.$$

Example 6.1 Computation of Nontrivial Solutions

Let $C : (\frac{2t}{1+t^2}, \frac{1-t^2}{1+t^2}, 0)$, and $\mathbf{n}(t) = (\frac{4t}{1+t^2}, \frac{2-2t^2}{1+t^2}, 0)$, which is from an intersection of a sphere $x^2 + y^2 + z^2 - 1 = 0$ with the plane $z = 0$. To find a surface of degree 2 which Hermite interpolates C , we let $f(x, y, z) = c_1x^2 + c_2y^2 + c_3z^2 + c_4xy + c_5yz + c_6zx + c_7x + c_8y + c_9z + c_{10}$. From the containment condition, we get 5 equations, $c_{10} - c_8 + c_2 = 0$, $2c_7 - 2c_4 = 0$, $2c_{10} - 2c_2 + 4c_1 = 0$, $2c_7 + 2c_4 = 0$, $c_{10} + c_8 + c_2 = 0$, and from the tangency condition, we also get 5 equations, $-2c_9 + 2c_5 = 0$,

$-4c_6 = 0, -4c_5 = 0, 4c_6 = 0, 2c_9 + 2c_5 = 0$. In a matrix form,

$$Ax = \begin{pmatrix} 0 & 1 & 0 & 0 & 0 & 0 & 0 & -1 & 0 & 1 \\ 0 & 0 & 0 & -2 & 0 & 0 & 2 & 0 & 0 & 0 \\ 4 & -2 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 2 \\ 0 & 0 & 0 & 2 & 0 & 0 & 2 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 \\ 0 & 0 & 0 & 0 & 2 & 0 & 0 & 0 & -2 & 0 \\ 0 & 0 & 0 & 0 & 0 & -4 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & -4 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 4 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 2 & 0 & 0 & 0 & 2 & 0 \end{pmatrix} \begin{pmatrix} c_1 \\ c_2 \\ c_3 \\ c_4 \\ c_5 \\ c_6 \\ c_7 \\ c_8 \\ c_9 \\ c_{10} \end{pmatrix} = \mathbf{0}.$$

The Σ in the SVD of A is $\text{diag}(5.65685, 4.89898, 4.89898, 2.82843, 2.82843, 2.82843, 2.0, 1.41421, 0.0, 0.0)$.² Hence, we see that the rank of A is 8, and the null space of A is

$$x = \tau_1 \cdot v_9 + \tau_2 \cdot v_{10} = \tau_1 \begin{pmatrix} 0.0 \\ 0.0 \\ 1.0 \\ 0.0 \\ 0.0 \\ 0.0 \\ 0.0 \\ 0.0 \\ 0.0 \\ 0.0 \end{pmatrix} + \tau_2 \begin{pmatrix} 0.57735 \\ 0.57735 \\ 0.0 \\ 0.0 \\ 0.0 \\ 0.0 \\ 0.0 \\ 0.0 \\ 0.0 \\ -0.57735 \end{pmatrix}.$$

The nontrivial solutions are obtained by making sure that the free parameters, τ_1 and τ_2 , do not vanish simultaneously. Hence, the Hermite interpolating surface is $f(x, y, z) = 0.57735\tau_2x^2 + 0.57735\tau_2y^2 + \tau_1z^2 - 0.57735\tau_2 = 0$ which has one degree of freedom in controlling its coefficients. $f(x, y, z) = 0$ can be made to contain a point, say, $(1, 0, 1)$. That is, $f(1, 0, 1) = 0.57735\tau_2 + \tau_1 - 0.57735\tau_2 = \tau_1 = 0$. So, the circular cylinder $f(x, y, z) = 0.57735\tau_2(x^2 + y^2 - 1) = 0$ is an appropriate Hermite interpolating surface. \square

6.2 Bounding the Degree of Surfaces

The total number of linear equations generated for a possible algebraic surface of degree n to Hermite interpolate k points with fixed constant "normal" directions and also to contain, with C^1 continuity, l space curves of degree d with assigned "normal" directions, varying as a polynomial of degree m , is $3k + (2n + m - 1)dl + 2l$. This number reduces to $3k + (2n - 1)dl + ml + 2l$ when all the space curves and associated "normals" are defined parametrically.

For a given configuration of points, curves, and "normals" data, the above interpolation scheme allows one to both upper and lower bound the degree of Hermite interpolating surfaces.

1. **Lower Bound** Let $k(n)$ be the rank of a homogeneous system of linear equations, derived from the given geometric configuration and surface degree n . The rank tells us the exact

²The subroutine dsydc of Linpack was used to compute the SVD of a matrix.

number of independent constraints on the coefficients of the desired algebraic surface of degree n . Dependencies arise from spatial interrelationships of the given points and curves. From the rank then, we can conclude that there exists no algebraic surface of a degree less than or equal to n_0 where n_0 is the largest n such that $K(n) < k(n)$ with $K(n) = \binom{n+3}{3} - 1$.

2. **Upper Bound** Alternatively, the smallest n can be chosen such that $K(n) \geq k(n)$, where, again, $K(n)$ is the number of independent unknown coefficients, and $k(n)$ is the rank of the linear system. The nontrivial solutions of the linear system represents a $(K - k)$ -parameter family of algebraic surfaces of degree n which interpolates the given geometric data. We then select suitable surfaces from this family, which additionally satisfy our nonsingularity and irreducibility constraints.³

One way to apply the Hermite interpolation technique to find a "lowest" degree algebraic surface which has given geometric properties, is to search through the degrees, i.e. from $n = 1, 2, 3, \dots$ for an interpolating surface. In Example 6.8 in Subsection 6.3, we illustrate how the rank could be conjectured *a priori*, without generating linear equations and then actually computing it, using only topological interrelationships between input curves and normal directions. However, since the dependencies between linear equations do depend on the specific spatial interrelationships of the given points and curves, it is, in general, quite difficult to bound the degree of interpolating surfaces *a priori*. (For example, it is possible to design input data, made of an arbitrary number of degree 4 curves with "normal" directions, which are interpolated by a single degree 2 implicit surface.)

We now enumerate some results which lower bound the degree of feasible Hermite interpolating surfaces.

1. Two skewed lines in space with constant-direction normals cannot be Hermite interpolated with quadric surfaces. The only quadric which satisfies both containment and tangency conditions reduces into two planes.
2. Two lines in space with constant-direction normals can be Hermite interpolated with a quadric surface if and only if the lines are parallel or intersect at a point, and the normals are not orthogonal to the plane containing them. The quadric is a "cylinder" when the lines are parallel, and a "cone" when the lines intersect.
3. The minimum degree of an algebraic surface, which Hermite interpolates two lines in space, one with a constant direction normal, the other with a linearly varying normal is three.
4. Two lines with linearly varying normals can be Hermite interpolated by a quadric in only some special cases. In general, a surface of at least degree three is needed. When quadric surface interpolation is possible, the quadric is either a hyperboloid of one sheet (the two lines may be parallel, intersecting, or skewed) or a hyperbolic paraboloid (the two lines can only be intersecting or skewed).

³However, some of these interpolating surfaces might still not be suitable for the design application they were intended to benefit. These problems arise when the given points or curves are smoothly interpolated, but, lie on separate real components of the same nonsingular, irreducible algebraic surface.

6.3 Examples

In this subsection, we exhibit the method of Hermite interpolation by constructing the lowest degree Hermite interpolating surfaces for “joining” and “blending” primary surfaces of solid models as well as for “fleshing” curved wire frame models of physical objects.

Example 6.2 (JOINING 1) *A Cubic Surface for Smoothly Joining Two Elliptic Cylinders*

Consider computing a lowest degree surface which can smoothly join two truncated elliptic cylinders $CYL_1 : (y + 1)^2 + \frac{z^2}{4} - 1 = 0$ for $x \leq -2$ and $CYL_2 : 25z^2 + 36y^2 - 96xy + 64x^2 - 100 = 0$ for $3x + 4y \geq 0$. Here, we illustrate the Hermite interpolation technique which not only computes the *unique* cubic interpolating surface but also proves that degree three is the *lowest* for an algebraic surface to satisfy the smooth-join requirement for this configuration. As before, we take a circle $C_1 : (-2, \frac{-2t^2}{1+t^2}, \frac{4t}{1+t^2})$ on CYL_1 with the associated rational “normal” $n_1(t) : (0, \frac{2-2t^2}{1+t^2}, \frac{2t}{1+t^2})$ and the circle $C_2 : (\frac{-4+4t^2}{5+5t^2}, \frac{3-3t^2}{5+5t^2}, \frac{4t}{1+t^2})$ on CYL_2 with the associated rational “normal” $n_2(t) : (\frac{-160+160t^2}{1+t^2}, \frac{120-120t^2}{1+t^2}, \frac{200t}{1+t^2})$. Both C_1 and C_2 's “normals” are respectively chosen in the same directions as the gradients of their corresponding containing surfaces CYL_1 and CYL_2 . This ensures that any Hermite interpolating surface for C_1 and C_2 will also meet CYL_1 and CYL_2 smoothly along these curves. A degree two algebraic surface does not suffice for Hermite interpolation, since the rank of the resulting linear system is greater than 9, the number of independent unknowns. (Note that a quadratic surface has 10 coefficients.) Next, as a possible Hermite interpolant, consider a degree three algebraic surface with 19 independent unknown coefficients. Applying the Hermite interpolation method of Subsection 4.2.2, to the curves results in 26 equations (28 equations are supposed to be generated, but 2 of the 28 are degenerate.). The rank of this linear system is 19, and thus there is a unique cubic Hermite interpolating surface, which is $f(x, y, z) = \tau_1(2yz^2 - xz^2 - 5z^2 + 8y^3 - 4xy^2 - 4y^2 + 8x^2y + 24xy - 8y - 4x^3 - 11x^2 + 4x + 20)$. See Figure 1. \square

Example 6.3 (JOINING 2) *A Quartic Surface for Smoothly Joining Three Circular Cylinders*

Consider computing 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 $y \geq 2$.

In [29], a degree 5 surface is found for joining these cylinders. After applying the Hermite interpolation algorithm, we find out that the minimum degree for such joining surface is 4, and we get a 2-parameter (one independent parameter) family of algebraic surfaces.

As before, we take 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” $n_1(t) : (\frac{4t}{1+t^2}, \frac{2-2t^2}{1+t^2}, 0)$, 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” $n_2(t) : (0, \frac{4t}{1+t^2}, \frac{2-2t^2}{1+t^2})$, 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” $n_3(t) : (\frac{4t}{1+t^2}, 0, \frac{2-2t^2}{1+t^2})$. 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 Hermite 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 for Hermite interpolation, since the rank of the resulting linear system is greater than 19, the number of independent unknowns. Next, as a possible Hermite interpolant, consider

a degree four algebraic surface with 34 independent unknown coefficients. Applying the Hermite interpolation method to the curves results in 52 equations. The rank of this linear system is 33, and thus there is a one independent parameter family of quartic Hermite interpolating surfaces, which is $f(x, y, z) = r_1 z^4 + \frac{r_2+10r_1}{12} yz^3 + \frac{r_2+10r_1}{12} xz^3 - \frac{r_2+10r_1}{3} z^3 + 2r_1 y^2 z^2 + \frac{r_2+10r_1}{12} xyz^2 - \frac{r_2+10r_1}{3} yz^2 + 2r_1 x^2 z^2 - \frac{r_2+10r_1}{3} xz^2 + r_2 z^2 + \frac{r_2+10r_1}{12} y^3 z + \frac{r_2+10r_1}{12} xy^2 z - \frac{r_2+10r_1}{3} y^2 z + \frac{r_2+10r_1}{12} x^2 yz - \frac{r_2+10r_1}{3} xyz + \frac{r_2+10r_1}{3} yz + \frac{r_2+10r_1}{12} x^3 z - \frac{r_2+10r_1}{3} x^2 z + \frac{r_2+10r_1}{4} xz + \frac{r_2+10r_1}{3} z + r_1 y^4 + \frac{r_2+10r_1}{12} xy^3 - \frac{r_2+10r_1}{3} y^3 + 2r_1 x^2 y^2 - \frac{r_2+10r_1}{3} xy^2 + r_2 y^2 + \frac{r_2+10r_1}{12} x^3 y - \frac{r_2+10r_1}{3} x^2 y + \frac{r_2+10r_1}{4} xy + \frac{r_2+10r_1}{3} y + r_1 x^4 - \frac{r_2+10r_1}{3} x^3 + r_2 x^2 + \frac{r_2+10r_1}{3} x + \frac{5r_1-7r_2}{3}$.

An instance of this family ($r_1 = 1, r_2 = 10$) is shown in Figure 2. It should be noted that every instance is not always appropriate for the geometric modeling purpose. The quartic surface on the left in Figure 3 is one used in Figure 2. On the other hand, the surface on the right, which is not useful in light of geometric modeling, is also in the same family with $r_1 = 1$ and $r_2 = -1$. \square

Example 6.4 (JOINING 3) A Quartic Surface for Smoothly Joining Four Circular Cylinders

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

The Hermite interpolation technique indicates that the minimum degree for such a joining surface is 4, and computes a 2-parameter (one independent parameter) family of algebraic surfaces which is $f(x, y, z) = \frac{r_1}{14} z^4 + \frac{r_1}{7} y^2 z^2 - \frac{4r_1}{7} yz^2 + r_1 z^2 + \frac{r_1}{14} y^4 - \frac{4r_1}{7} y^3 + r_1 y^2 + \frac{4}{7} r_1 y + \frac{14r_2+15r_1}{224} x^4 - \frac{14r_2+15r_1}{28} x^2 + r_2$. An instance of this family ($r_1 = 392, r_2 = -868$) is shown in Figure 4. \square

Example 6.5 (BLENDING 1) A Hyperboloid Patch for Blending Two Circular Cylinders

The case of two circular cylinders is a common test case for "blending" algorithms. Various different ways have been given, (for example, see [16, 20, 29]) for computing a suitable surface which "smoothes" or "blends" the intersection of two equal radius cylinders, $CYL_1 : x^2 + z^2 - 1 = 0$ and $CYL_2 : y^2 + z^2 - 1 = 0$. We consider an ellipse C_1 on CYL_1 (it is the intersection with the plane $3x + y = 0$), defined parametrically, $C_1 : (\frac{2t}{1+t^2}, \frac{-6t}{1+t^2}, \frac{1-t^2}{1+t^2})$ with the associated rational "normal" $n_1(t) = (\frac{4t}{1+t^2}, 0, \frac{2-2t^2}{1+t^2})$, and the ellipse C_2 on CYL_2 defined implicitly, $C_2 : ((y^2 + z^2 - 1 = 0, x + 3y = 0)$ with the associated "normal" $n_2(x, y, z) = (0, 2y, 2z)$. As a possible Hermite interpolant, we consider a degree two algebraic surface. Applying the method of Subsection 4.2.2, to C_1 results in 8 equations, 5 from the containment condition and 3 from the tangency condition. (5 equations are supposed to be generated, but 2 of these turn out to be degenerate). For C_2 , we use the method of Subsection 4.2.1, and first compute $L_c = \{(0, 0, 1), (-3, 1, 0), (3, -1, 0), (-2.4, 0.8, -0.6), (2.4, -0.8, -0.6)\}$ and $L_t = \{[(0, 0, 1), (0, 0, 2)], [(-3, 1, 0), (0, 2, 0)], [(3, -1, 0), (0, -2, 0)], [(-2.4, 0.8, -0.6), (0, 1.6, -1.2)], [(2.4, -0.8, -0.6), (0, -1.6, -1.2)]\}$. From these lists, we get 10 equations, 5 from the containment condition and another 5 from the tangency condition. Hence, overall the linear system consists of 9 independent unknowns and 18 equations. The rank of this system is 9, and hence we get the unique surface solution $f_1(x, y, z) = r_1(x^2 + y^2 - 8z^2 + 6xy + 8 = 0)$. This quadric satisfies both the nonsingularity and irreducibility constraints. It is a hyperboloid of one sheet and the lowest degree surface which "blends", together with a symmetric hyperboloid $f_2(x, y, z) = r_1(x^2 + y^2 - 8z^2 - 6xy + 8 = 0)$, the intersection of the two cylinders. See Figure 5. \square

Example 6.6 (BLENDING 2) A Quartic Surface for Blending Two Elliptic Cylinders

In this example, we compute the lowest degree surface which blends two perpendicular elliptic cylinders. We have seen a quadric blending of the circular cylinders in Example 6.5. Here, we try a quartic blending surface by taking different types of input curves.

Input to Hermite interpolation is defined by $CYL_1 : y^2 + 4z^2 - 4 = 0$ for $x \geq 1$, $CYL_2 : y^2 + 4z^2 - 4 = 0$ for $x \leq -1$, $CYL_3 : 9x^2 + y^2 - 9 = 0$ for $z \geq 1$, and $CYL_4 : 9x^2 + y^2 - 9 = 0$ for $z \leq -1$.

The Hermite interpolation algorithm proves that 4 is the minimum degree for such a blending surface, and generates a linear system with 72 equations of rank 33. The 2-parameter (one independent parameter) family of algebraic surfaces is $f(x, y, z) = r_1 z^4 - \frac{8r_2 + 81r_1}{72} y^2 z^2 - \frac{8r_2 + 81r_1}{8} x^2 z^2 + \frac{8r_2 + 65r_1}{8} z^2 - \frac{8r_2 + 99r_1}{288} y^4 - \frac{8r_2 + 81r_1}{32} x^2 y^2 + \frac{104r_2 + 1053r_1}{288} y^2 + \frac{81r_1}{16} x^4 + r_2 x^2 - \frac{16r_2 + 65r_1}{16}$. An instance of this family ($r_1 = 1$, $r_2 = 2$) is shown in Figure 6. \square

Example 6.7 (FLESHING 1) A Quartic Surface for Fleshing a Circular Wireframe

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 7, $face_1$ and $face_3$ are filled with the patches taken from the sphere $x^2 + y^2 + z^2 - 25 = 0$ (green patches). To smoothly flesh the remaining faces requires degree 4 surface patches. Applying the Hermite interpolation method in either Subsection 4.2.2 or Subsection 4.2.1 to C_1 and C_2 results in 11-parameter (10 independent) family of quartic interpolating surfaces, which is $f(x, y, z) = r_1 z^4 + (r_2 y + r_6 x + 5r_4) z^3 + (r_3 y^2 + (r_7 x + 5r_8) y + r_{10} x^2 + 5r_{11} x - 25r_9 - 25r_1) z^2 + (r_2 y^3 + (r_6 x + 5r_4) y^2 + (r_2 x^2 - 25r_2) y + r_6 x^3 + 5r_4 x^2 - 25r_6 x - 125r_4) z + (r_3 - r_1) y^4 + (r_7 x + 5r_8) y^3 + (r_5 x^2 + 5r_{11} x - 25r_9 - 25r_3 + 25r_1) y^2 + (r_7 x^3 + 5r_8 x^2 - 25r_7 x - 125r_8) y + (r_{10} - r_1) x^4 + 5r_{11} x^3 + (-25r_9 - 25r_{10} + 25r_1) x^2 - 125r_{11} x + 625r_9$. An instance $f(x, y, z) = -1250 - x^4 - y^4 - x^2 z^2 - y^2 z^2 + 50z^2 + 75y^2 + 75x^2$ of this family is used to flesh $face_2$ and $face_4$ in Figure 7 (red patches). \square

Example 6.8 (FLESHING 2) A Quintic Surface for Fleshing a Triangular Wireframe

Figure 8 shows an instance of a 5-parameter family of quintic surfaces which fleshes the triangular wireframe, made of three conic curves with quadratic normals (both curves and normals are in quadratic rational parametric form) where the three curves meet pairwise, and the normals coincide at the three intersection points. We are currently working on a polyhedron-smoothing scheme in which this kind of triples of quadratic curve-normal pairs are automatically generated.

Consider the problem of smooth fleshing of the three curves with a degree n algebraic surface $f(x, y, z) = 0$. In order for f to contain a curve that is a quadratic rational parametric curve, $2n + 1$ linear equations are generated. Also, in order for f to contain, smoothly, a curve which has a normal in a quadratic rational parametric form, $2(n - 1) + 1$ additional linear equations should be satisfied. Since there are 3 such curves, $3\{(2n + 1) + (2(n - 1) + 1)\} = 12n$ equations are produced.

On the other hand, we observe some geometric dependency between the curves which lead to algebraic dependency. First, since the curves intersect pairwise, there must be three rank deficiencies between the equations from containment conditions.⁴ Secondly, at each intersection point,

⁴If we always choose the intersection points for the list L_c of each curve in the algorithm in Section 4.2.1, three equations are generated twice.

two incident curves automatically determine the normal at the point. This means satisfying containment conditions for the 3 curves guarantees that any interpolating surface has normals at the three points as specified. This fact implies that, for each curve, there are two rank deficiencies between the linear equations for containment condition, and the equations for its tangency condition.⁵ Hence, 6 additional rank deficiencies with the previous 3 indicate that $12n - 9$ is the maximum possible rank of the linear system.

Since $f(x, y, z) = 0$ of degree n has $\binom{n+3}{3}$ coefficients, and the rank of the linear system should be less than the number of coefficients in order for a nontrivial surface to exist, we see that 5 is the minimum degree. In the quintic surface case, there are 56 coefficients and the rank is 51, which results in a family of interpolating surfaces with at least 4 degrees of freedom. Even though some special combination of three conics with quadratic normals can be interpolated by a surface of degree less than 5, the probability that such spatial dependency occurs, given an arbitrary triple of conics with normals, is infinitesimal. Hence, we can say that 5 is the minimum degree required *with a probability one*. As the above counting implies, there *always* exist quintic surfaces which smoothly interpolate the three given curves. Unfortunately, this counting above does not prove there exists an interpolant which meets all the geometric shape and design requirements in a specific application. Achieving all these properties simultaneously remains a topic of further research. In the next section, we present an interactive technique of controlling the shape of a suitable surface instance selected from a large family of interpolating surfaces. \square

7 Interactive Shape Control of Hermite Interpolating Surfaces

The result of a Hermite 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 $\mathbf{r} = r_1, r_2, \dots, r_p$). The equation for the family has the form

$$f(x, y, z) = \sum_{i=0}^n \sum_{j=0}^{n-i} \sum_{k=0}^{n-i-j} c_{ijk} \cdot x^i y^j z^k = 0 \quad (9)$$

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. In this section, we 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 $w = f(x, y, z)$. Of course, since we consider 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 our purpose, 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

⁵Again, for each curve, we can choose point-normal pairs at the two end points. The resulting two linear equations should be linearly dependent on the equations for containment.

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 (9), which is a linear (not necessarily homogeneous) combination of r . All this together defines the p -parameter algebraic surface family in Barycentric coordinates,

$$F(s, t, u) = \sum_{i=0}^n \sum_{j=0}^{n-i} \sum_{k=0}^{n-i-j} w_{ijk} \cdot \binom{n}{i, j, k} \cdot s^i t^j u^k (1-s-t-u)^{n-i-j-k} = 0 \quad (10)$$

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.

Example 7.1 Conversion from Power to Bernstein

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 (9) with $n = 2$ and where $c_{200} = r_1$, $c_{110} = 2r_1$, $c_{101} = r_4$, $c_{100} = -2r_1$, $c_{020} = r_1$, $c_{011} = r_5$, $c_{010} = -2r_1$, $c_{002} = r_3$, $c_{001} = r_2$, 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 (10) 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 $w_{200} = r_1$. \square

As $F(s, t, u) = 0$ in (10), 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, : $w_1 - w_2 - w_3 - 6 = 0$. (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

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

from the linear weight expressions

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

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

This is easily achieved by Gaussian elimination. Changing some weights can now be considered as moving from a weight vector $\mathbf{W} = (w_1, w_2, \dots, w_c)$ to another $\mathbf{W}' = (w'_1, w'_2, \dots, w'_c)$, with the constraint that $\Delta\mathbf{W} = \mathbf{W}' - \mathbf{W}$ is a solution of the computed system of invariant equations.

Example 7.2 Shape Control of a Family of Quadric Surfaces

The invariant equations for weight change of a surface in Example 7.1 are $\Delta w_{010} + \Delta w_{000} = 0$, $\Delta w_{020} - \Delta w_{000} = 0$, $\Delta w_{100} + \Delta w_{000} = 0$, $\Delta w_{110} - \Delta w_{000} = 0$, $\Delta w_{200} - \Delta w_{000} = 0$. Figure 9 (upper left) shows an instance of a family in Example 7.1 where $w_{000} = -4$, $w_{001} = 4$, $w_{002} = 8$, $w_{010} = 4$, $w_{011} = 14$, $w_{020} = -4$, $w_{100} = 4$, $w_{101} = 12$, $w_{110} = -4$, and $w_{200} = -4$.

Now, consider the control point P_{002} (the red point). Suppose a surface designer decides to pull the surface patch toward P_{002} . This can be achieved by decreasing the value of w_{002} , say, $\Delta w_{002} = -7$. (See [23, 24] for more introduction to the relation between the surface shape and weights of control points.)

Other Δw_{ijk} can be arbitrarily chosen as long as they satisfy the invariant equations. Let $\Delta w_{000} = \Delta w_{200} = \Delta w_{110} = \Delta w_{020} = -1$, $\Delta w_{100} = \Delta w_{010} = 1$, $\Delta w_{001} = -4$, $\Delta w_{101} = \Delta w_{011} = -2$. The new instance is shown in Figure 9 (upper right). \square

Example 7.3 Shape Control of a Family of Quartic Surfaces

Figure 9 (bottom left) illustrates three different instances of $f(x, y, z) = 0$ in Example 6.3 obtained by changing the value of w_{000} . Each time w_{000} is varied, the invariant equations are met, and each instance surface Hermite \mathbf{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 $w_{000} > 0$. \square

Now, 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 ($i = 2, 3, \dots, c$). 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 $(\Delta w_2, \dots, \Delta w_c)$, and hence the 2-norm of $\Delta\mathbf{W}$. Note that $\|\Delta\mathbf{W}\|_2^2 = \Delta w_2^2 + \Delta w_3^2 + \dots + \Delta w_c^2$. For a change $\Delta w_1 = d$, we see that

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

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

In order to minimize the norm of $\Delta \mathbf{W}_0$, the quadratic function $Q(q_1, q_2, \dots, q_{p-1})$ needs to be minimized. Since Q is quadratic, the minimum solution can be obtained straightforwardly by solving the linear system $\nabla Q(q_1, q_2, \dots, q_{p-1}) = \mathbf{0}$. If the (unique) minimum solution point is $Q^0 = (q_1^0, q_2^0, \dots, q_{p-1}^0)$, then $\Delta \mathbf{W}^0 = (d, \Delta w_2^0, \dots, \Delta w_c^0)$ corresponding to Q^0 will define the desired change of weights of w_2, \dots, 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 $\Delta w_1 = d$.

Example 7.4 *Heuristic Approach to Shape Control using a weighted 2-norm*

Consider the surface in Example 7.2 again. This time the surface designer wishes to pull the patch more toward, say, P_{002} , and hence sets $\Delta w_{002} = -15$. From the invariant equations in which Δw_{002} is replaced by -15, $\Delta w_{000} = \Delta w_{020} = \Delta w_{110} = \Delta w_{200} = s_1$, $\Delta w_{010} = \Delta w_{100} = -s_1$, $\Delta w_{001} = s_2$, $\Delta w_{101} = s_3$, $\Delta w_{011} = s_4$ one obtains the quadratic function $Q(s_1, s_2, s_3, s_4) = 225 + 6s_1^2 + s_2^2 + s_3^2 + s_4^2$. The quadric function Q has its global minimum for $s_1 = s_2 = s_3 = s_4 = 0$. Hence, the influence of the change in all the weights except w_{002} , is minimized by setting their Δw values 0, i.e. not changing them at all.

This new instance is shown in Figure 9 (bottom right). Note that the overall shape of the new surface patch, other than close to P_{002} , has not changed as much as the surface patch in Figure 9 (upper right), even though here w_{002} has been decreased by a larger amount. \square

8 Concluding Remarks

We have implemented the Hermite interpolation algorithm, as presented in Section 3 and 4. The program takes as input any collection of points and curves, with/without associated "normals". Both implicit and rational parametric representations of the space curves and "normals" are allowed. The homogeneous linear system of equations generated by the algorithm is decomposed using the singular value decomposition method. As a result, the rank of the system, and the null space (that is, the solution of the system) are computed. The nontrivial solutions (and hence the coefficients of the corresponding surfaces), if any, are expressed in terms of linear combinations of free parameters and represent a family of interpolation surfaces.

To select a suitable interpolating surface from the computed family, requires an assignment of values to the free parameters. This is achieved by first converting the solution polynomial representation of the interpolating family to polynomials over a tetrahedral Bernstein basis together with weight invariants for the weights on each vertex of the control tetrahedra. Next, an initial assignment of weight values (consistent with the weight invariants) is chosen and then interactively modified to yield a desirable interpolating solution surface. The final weight values in this procedure yield the specific assignment of values to free parameters.

Even though the change of polynomial representations from power to Bernstein bases yields additional geometry to the coefficients of the polynomial representation of the algebraic surface, for large parameter (four and greater) family of surfaces, much depends on the initial assignment of weight values. One possibility that we are currently exploring for solution instantiation from

large parameter families is the use of least-square approximation [6] for implicit algebraic surfaces to compute these initial vector of values. Additional data points, curves, and even simple surfaces are inserted by the designer, in near proximity to the desired surface and the free parameters of the interpolating solution family are chosen to minimize the square of the distance error. This overall problem of Hermite interpolation and least-squares approximation amounts to solving an optimization problem of the form :

$$\begin{aligned} & \text{minimize} && \mathbf{x}^T \mathbf{M}_A^T \mathbf{M}_A \mathbf{x} \\ & \text{subject to} && \mathbf{M}_I \mathbf{x} = \mathbf{z}, \\ & && \mathbf{x}^T \mathbf{M}_N \mathbf{x} = 1, \end{aligned}$$

where $\mathbf{M}_A, \mathbf{M}_I, \mathbf{M}_N$ are, respectively, matrices for least-squares approximation, Hermite interpolation and normalization. The solution can be obtained by computing eigenvectors of some reduced matrix. Details are provided in [6].

The triangular quintic surface in Example 6.8 was produced by applying least-squares approximation to the resulting 5-parameter family of interpolatory algebraic surfaces and some additional geometric data. In most of our examples, the least-squares approach proved to be an effective method to choose proper instances from a family of interpolatory algebraic surfaces. We are currently using this method to instantiate 5-parameter triangular quintic surface patches for construction of a compact implicit model of a human head, reconstructed from NMR imaging data.

There still remain several open problems which need to be solved before the entire class of algebraic surfaces can be effectively used for geometric design. First, as illustrated in Example 6.3, some of the irreducible interpolating algebraic surfaces produced directly from Hermite interpolation may not be suitable for the geometric design application they were initially intended to benefit. More precisely, the input data when interpolated may however lie on two separate real components of the algebraic surface. One heuristic, which we currently used is to interactively include additional points and curves to effectively bridge the gap between separate real components. However, the question remains open for a priori generating *polynomial constraints* on the coefficients of the interpolating surfaces, which would ensure that all given points and curves lie on the same continuous real surface component.

Secondly, polynomial constraints are again required to ensure that a specific solution patch of an algebraic surface has no singularities. Having some singularities is not always an unfavorable feature in the design of algebraic surfaces. For example, the blending surface in Example 6.6 is singular at the four points where the four cylinder contact each other. (In fact, the only solution for satisfying the "normals" along the four ellipses is to have singular points.) While singularities along the boundary of a patch may be allowable, no singularity would seem tolerable in the interior of the patch. Hence, deriving uniform polynomial constraints imposed on the family of interpolating solution surfaces which ensure nonsingularity of the solution surface patch, would be highly desirable.

Acknowledgments We wish to thank Bill Bouma who enhanced the C. Kolb's ray tracer adding capabilities for rendering arbitrary implicit surfaces, and helped us in generating and displaying some of the pictures on HP9000/370SRX.

References

- [1] S.S. Abhyankar. *Algebraic Space Curves*. Univ. of Montreal, 1970. Montreal Lecture Notes.
- [2] V. Anupam, C. Bajaj, and S. Klinkner. The SHILP solid modeling and display toolkit: A user's manual. Technical Report CSD-TR-988, Computer Sci.,Purdue, June 1990.
- [3] D. Arnon, G. Collins, and S. McCallum. Cylindrical algebraic decomposition 1: The basic algorithm. *Siam J. Computing*, 13(4):865–889, 1984.
- [4] C. Bajaj. Geometric modeling with algebraic surfaces. In D. Handscomb, editor, *The Mathematics of Surfaces III*, pages 3–48. Oxford Univ. Press, 1988.
- [5] C. Bajaj, C. Hoffmann, J. Hopcroft, and R. Lynch. Tracing surface intersections. *Computer Aided Geometric Design*, 5:285–307, 1988.
- [6] C. Bajaj, I. Ihm, and J. Warren. Exact and least-squares approximate g^k -fitting of implicit algebraic surfaces. Technical Report CSD-TR-944, Computer Sci.,Purdue, Jan. 1990.
- [7] R. Barnhill. Surfaces in CAGD : A survey with new results. *Computer Aided Geometric Design*, 2:1–17, 1985.
- [8] J. Bloomenthal. Polygonization of implicit surfaces. *Computer Aided Geometric Design*, 5:341–355, 1988.
- [9] J. Bloomenthal and B. Wyvill. Interactive techniques for implicit modeling. *Proc. of the 1990 Symposium on Interactive 3D Graphics*, pages 109–116, 1990.
- [10] W. Boehm, G. Farin, and J. Kahmann. A survey of curve and surface methods in CAGD. *Computer Aided Geometric Design*, 1:1–60, 1984.
- [11] A.D. DeRose. *Geometric Continuity : A Parametrization Independent Measure of Continuity for Computer Aided Geometric Design*. PhD thesis, Computer Science, Univ. of California, Berkeley, 1985.
- [12] G. Farin. Triangular Bernstein-Bezier patches. *Computer Aided Geometric Design*, 3:83–127, 1986.
- [13] R.T. Farouki. The approximation of nondegenerate offset surfaces. *Computer Aided Geometric Design*, 3:15–43, 1988.
- [14] G.E. Golub and C.F. Van Loan. *Matrix Computation*. The Johns Hopkins Univ. Press, Baltimore, MD, 1983.
- [15] P. Hanrahan. Ray tracing algebraic surfaces. *Computer Graphics : SIGGRAPH'83 Conference Proceedings*, 17(3):83–90, 1983.
- [16] C. Hoffmann and J. Hopcroft. Quadratic blending surfaces. *Computer Aided Design*, 18(6):301–306, 1986.

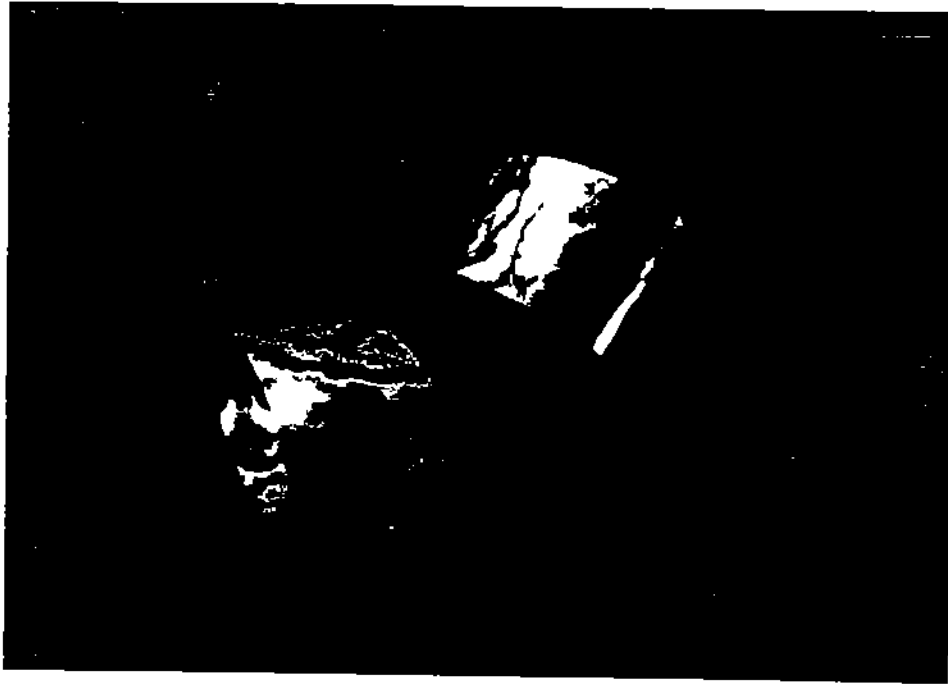


Figure 1: Smooth Joining of Two Cylinders with a Cubic Surface

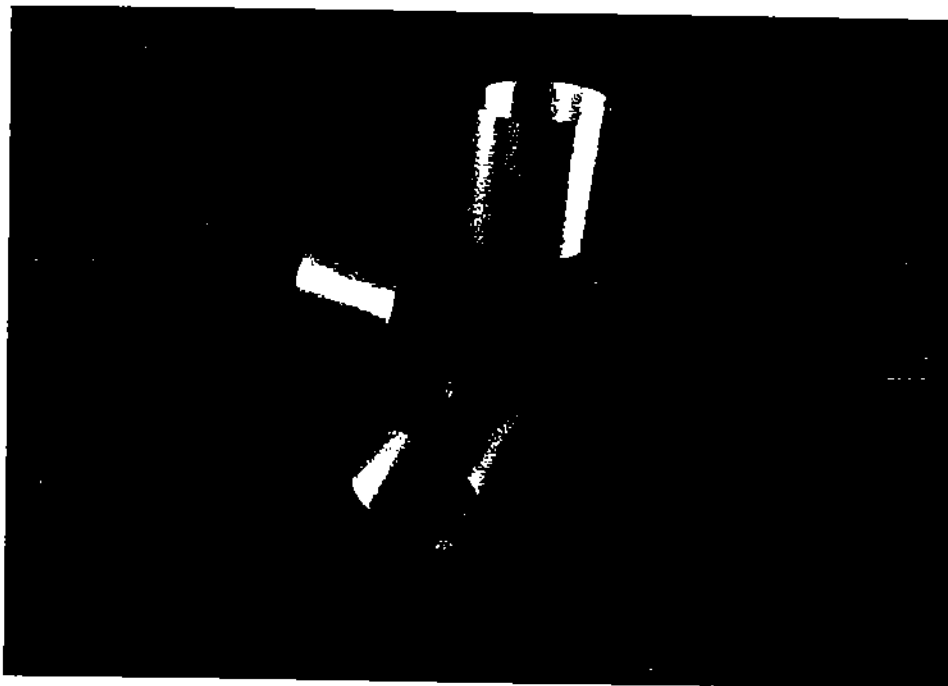


Figure 2: Smooth Joining of Three Cylinders with a Quartic Surface

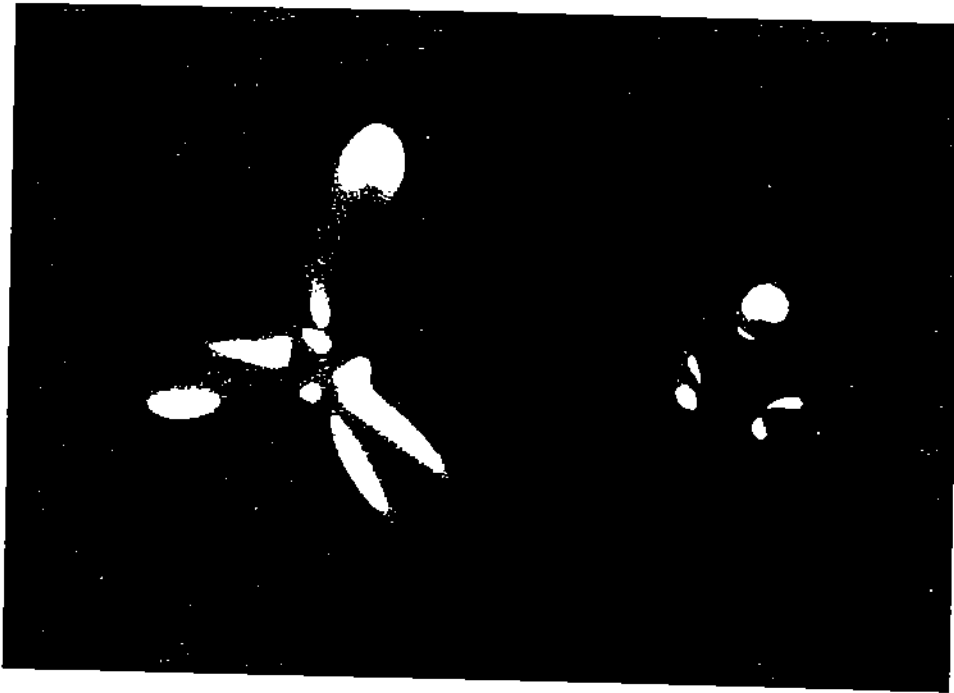


Figure 3: Two Instance Surfaces for Joining Three Cylinders

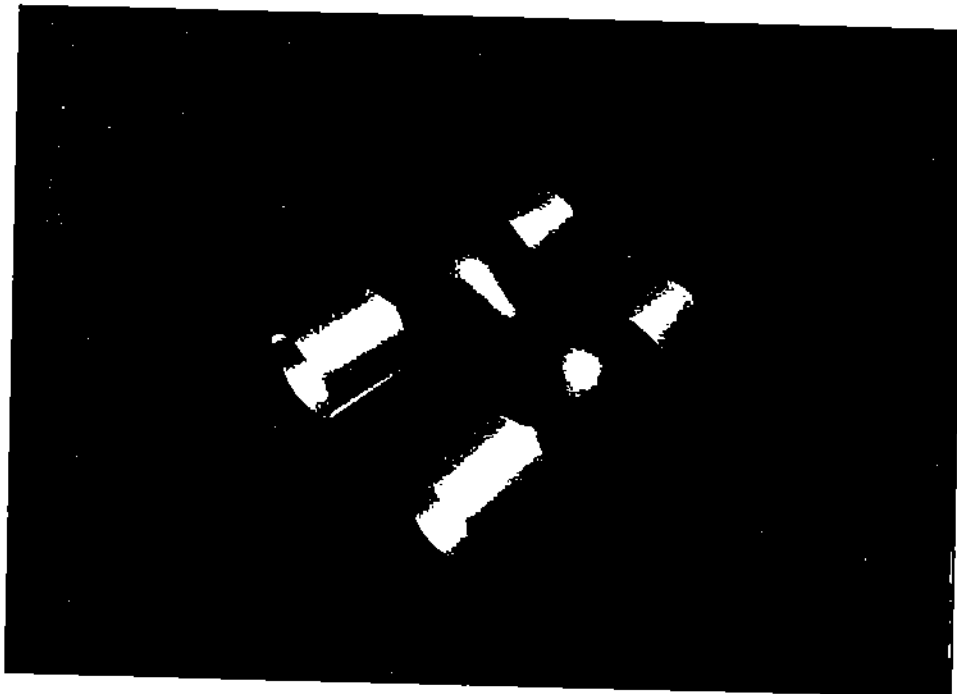


Figure 4: Smooth Joining of Four Cylinders with a Quartic Surface

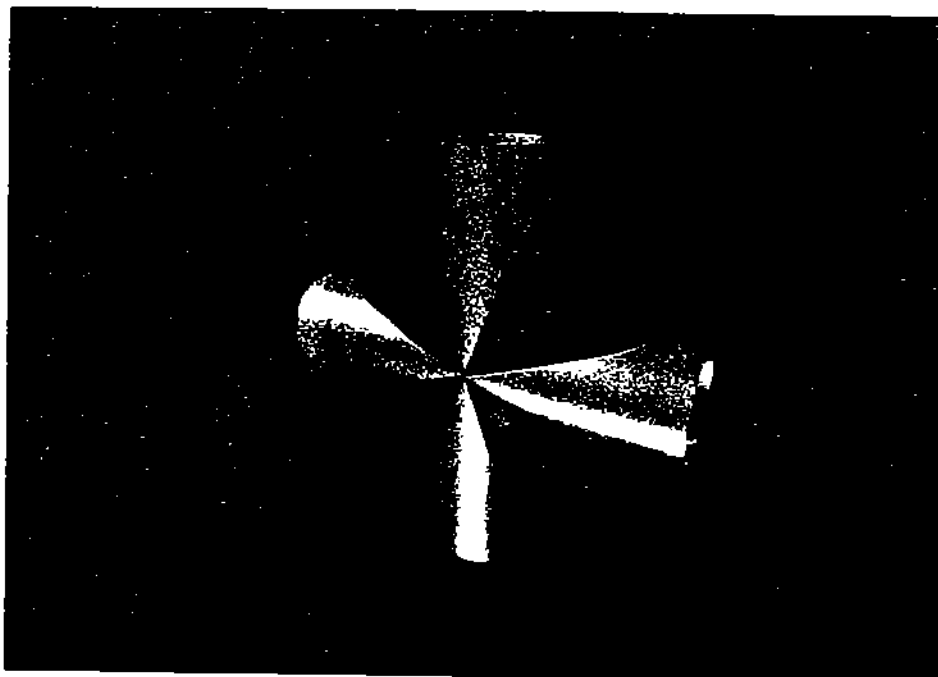


Figure 5: Smooth Blending of Two Cylinders with a Quadric Surface

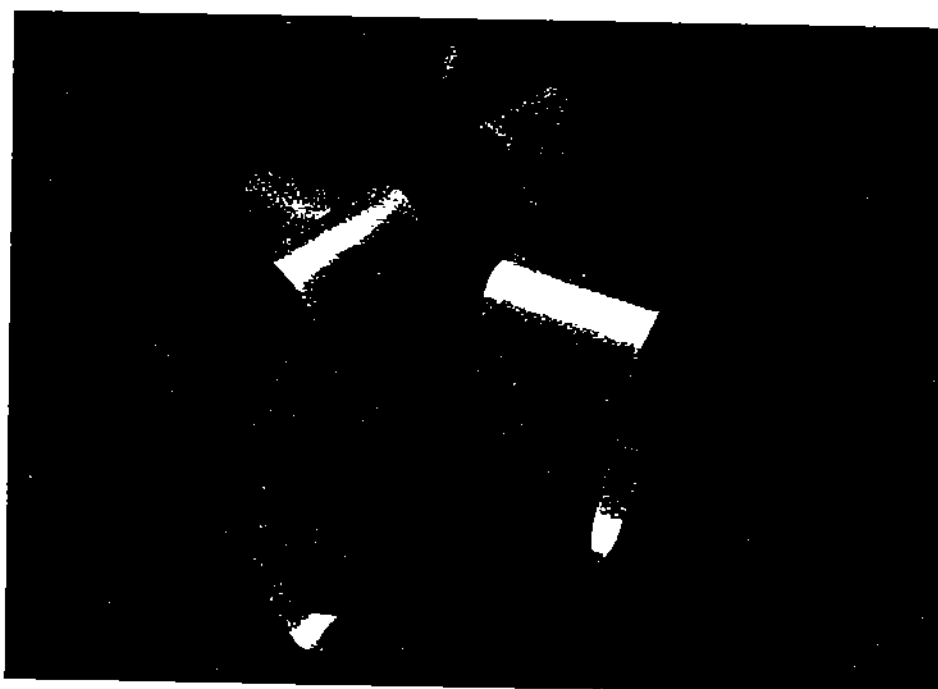


Figure 6: Smooth Blending of Two Cylinders with a Quartic Surface

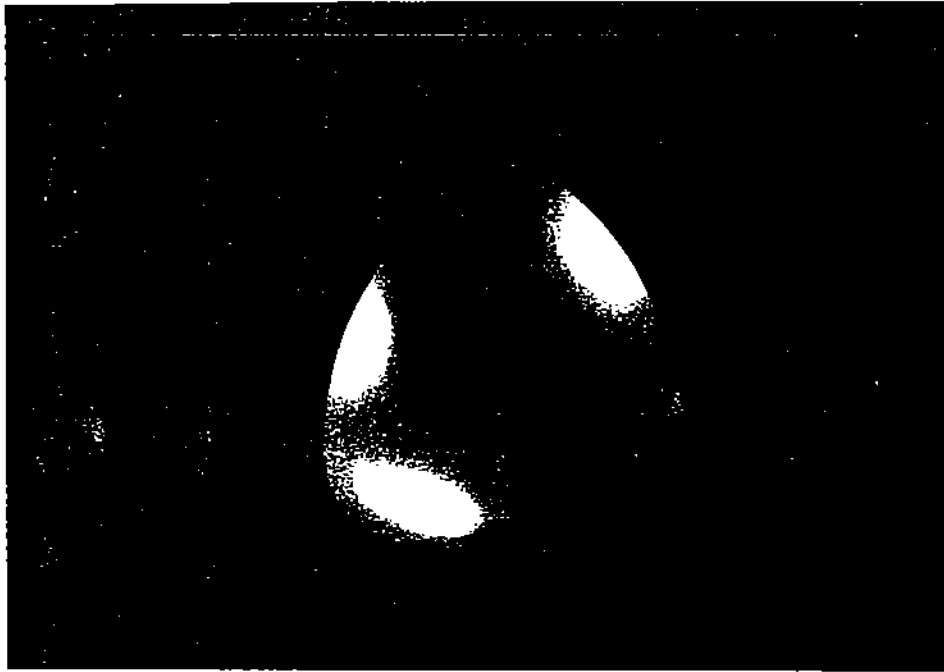


Figure 7: Smooth Fleshing of a Wireframe with Quadric and Quartic Surfaces

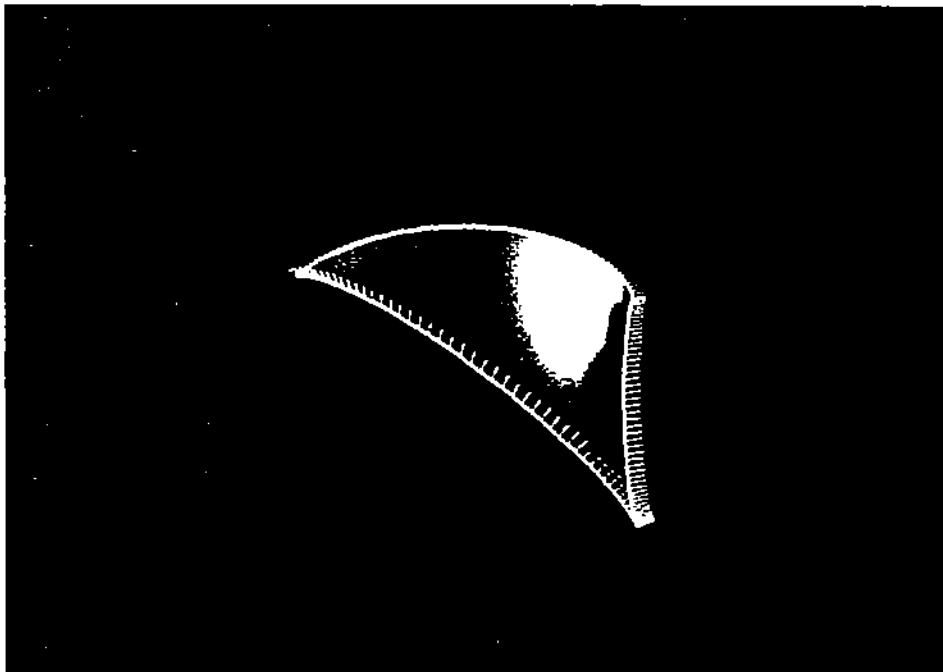


Figure 8: Smooth Fleshing of a Triangular Wireframe with a Quintic Surface

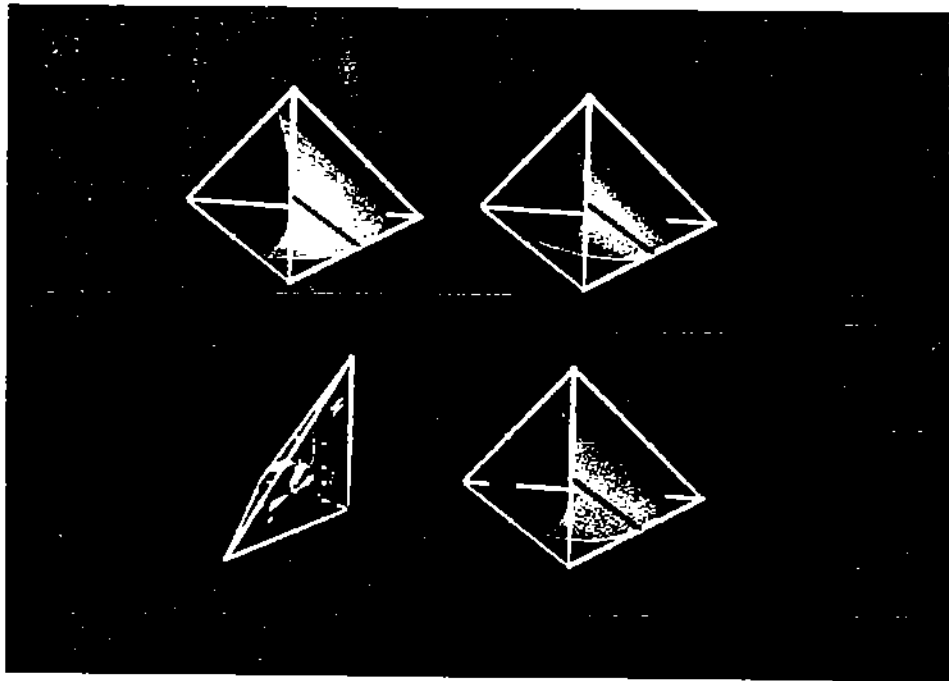


Figure 9: Interactive Shape Control