Higher-Order Interpolation and Least-Squares Approximation Using Implicit Algebraic Surfaces

CHANDRAJIT BAJAJ
Purdue University
and
INSUNG IHM
Sogang University
and
JOE WARREN
Rice University

In this article, we characterize the solution space of low-degree, implicitly defined, algebraic surfaces which interpolate and/or least-squares approximate a collection of scattered point and curve data in three-dimensional space. The problem of higher-order interpolation and least-squares approximation with algebraic surfaces under a proper normalization reduces to a quadratic minimization problem with elegant and easily expressible solutions. We have implemented our algebraic surface-fitting algorithms, and included them in the distributed and collaborative geometric environment SHASTRA. Several examples are given to illustrate how our algorithms are applied to algebraic surface design.

Categories and Subject Descriptors: F.2.1 [**Analysis of Algorithms and Problem Complexity**]: Numerical Algorithms and Problems—*computations on polynomials*; G.1.1 [**Numerical Analysis**]: Interpolation—*interpolation formulas*; G.1.2 [**Numerical Analysis**]: Approximation—*least squares approximation*; G.1.6 [**Numerical Analysis**]: Optimization—*constrained optimization*; I.3.5 [**Computer Graphics**]: Computational Geometry and Object Modeling—*curve, surface, solid, and object representation*

General Terms: Algorithms

Additional Key Words and Phrases: Algebraic surface, computer-aided geometric design, constrained quadratic optimization, distributed geometric-design environment, geometric continuity

1. INTRODUCTION

Computer-Aided Geometric Design (CAGD) deals with the representation and approximation of three-dimensional physical objects. A major task of CAGD is to automate the design process of such objects as car bodies,

C. Bajaj was supported in part by NSF grants CCR 90-02228, DMS 91-01424, and AFOSR contract 91-0276. I. Ihm was supported in part by David Ross Fellowship from Purdue University and J. Warren was supported in part by NSF grant IRI 88-10747.
Authors' addresses: C. Bajaj, Department of Computer Sciences, Purdue University, West Lafayette, IN 47907; I. Ihm, Department of Computer Science, Sogang University, Seoul, Korea; J. Warren, Department of Computer Science, Rice University, Houston, TX 77251.

airplane wings, and propeller blades, usually represented by smooth meshes of curves and surfaces. Research in surface design has been largely dominated by the theory of parametric curves and surfaces due to their highly desirable properties for trimmed surface design and computer graphics. In recent years, however, increasing attention has been paid to geometric design with implicitly defined algebraic curves and surfaces which provide a more comprehensive class of flexible surfaces, especially at lower degree. See Bajaj [1988; 1992; 1993], Bajaj and Ihm [1992a, 1992b], Bajaj et al. [1988], Bloomenthal [1988], Farouki [1988], Hoffmann and Hopcroft [1987], Owen and Rockwood [1987], Sederberg [1982; 1990a; 1990b], and Warren [1986; 1989].

An *algebraic surface* S in $\mathbf{R}^3$ is implicitly defined by a single polynomial equation $f(x, y, z) = 0$, whose coefficients are over the real numbers. The class of algebraic surfaces provides enough generality for geometric modeling as well as having the advantage of closure under several geometric operations like intersection, offsets, etc. [Bajaj 1988; 1993]. Smooth algebraic surfaces naturally define half spaces, which is a desirable property in solid modeling. Also, they are amenable to ray-surface intersection computation. A primary motivation for our work using implicit algebraic surfaces is based on the observation that implicitly represented algebraic surfaces are very natural for interpolation and least-squares approximation to both points and space curves with or without higher-order derivative information [Bajaj 1992; 1993]. As shown in this article, this fact yields compact computational schemes for a wide range of surface-fitting applications.

Fitting of algebraic curves (primarily lines and conics) has been considered by some authors [Albano 1974; Bajaj and Xu 1992; Bookstein 1979; Gnanadesikan 1977; Sampson 1982]. A good exposition of exact fits and least-squares fits of algebraic curves and surfaces through given data points is presented in Pratt [1987]. Sederberg [1990a] presents the idea of C^0 interpolation of data points and curves with algebraic surfaces. This previous work on interpolation is extended by Bajaj and Ihm [1992a], where they present algorithms for C^1 interpolation of points and space curves, represented either implicitly as the common intersection of algebraic surfaces or in the rational parametric form, possibly associated with tangential information.

In this article we present a computational model for low-degree, implicitly defined algebraic surface fitting. We consider least-squares fitting (approximation) as well as exact fitting (interpolation). This fitting scheme uses a proper normalization of coefficients of algebraic surfaces. The mathematical model we derive is a constrained minimization problem of the form:

$$\begin{aligned} &\textit{minimize} && \mathbf{x}^T \mathbf{M}_{\mathbf{A}}^T \mathbf{M}_{\mathbf{A}} \mathbf{x} \\ &\textit{subject to} && \mathbf{M}_{\mathbf{I}} \mathbf{x} = \mathbf{0}, \\ & && \mathbf{x}^T \mathbf{x} = 1, \end{aligned}$$

where $\mathbf{M}_{\mathbf{I}}$ and $\mathbf{M}_{\mathbf{A}}$ are matrices for interpolation and least-squares approximation, respectively, and $\mathbf{x}$ is a vector containing coefficients of an algebraic surface. In Section 2, we consider interpolation, least-squares approximation, and normalization in detail, and we show how the minimization problem is derived. Then, in Section 3, compact computational algorithms are explained with examples in algebraic surface rounding, blending, joining, and meshing. Finally, we discuss implementation and open problems for algebraic surface design in Section 4.

2. COMPUTATION OF MATRICES FOR ALGEBRAIC SURFACE DESIGN

2.1 Interpolation

2.1.1 *C^1 Interpolation for Implicit and Parametric Data.* Bajaj and Ihm [1992a] present a constructive characterization, called *Hermite interpolation*, of implicitly defined algebraic surfaces which *smoothly contain* given points and algebraic space curves, with associated *normal* directions, where these geometric input data are expressed either *implicitly* or *parametrically*. For an algebraic surface S: $f(x, y, z) = 0$ of degree n, the Hermite interpolation algorithm takes as input positional and first-derivative constraints on points and space curves and produces a homogeneous linear system $\mathbf{M_I x} = \mathbf{0}$, $\mathbf{M_I} \in \mathbf{R}^{n_r \times n_v}$, where $\mathbf{x}$ is a vector of the $n_v\left(= \binom{n+3}{3}\right)$ coefficients of S.[1] Only when the rank r of $\mathbf{M_I}$ is less than the number of the coefficients n_v does there exist a nontrivial solution to the system. All vectors except $\mathbf{0}$ in the nullspace of $\mathbf{M_I}$ form a family of algebraic surfaces, satisfying the given input constraints, whose coefficients are expressed by homogeneous combinations of q free parameters, where $q = n_v - r$ is the dimension of the nullspace.

Example 2.1 (*Generation of a Homogeneous Linear System*). Let $C(t) = ((2t/1 + t^2), (1 - t^2/1 + t^2), 0)$ be a quadratic curve with an associated normal vector $\mathbf{n}(t) = ((4t/1 + t^2), (2 - 2t^2/1 + t^2), 0)$. (This curve and normal direction are from the intersection of the sphere $x^2 + y^2 + z^2 - 1 = 0$ with the plane $z = 0$.) To find a quadratic surface $f(x, y, z) = c_1 x^2 + c_2 y^2 + c_3 z^2 + c_4 xy + c_5 yz + c_6 zx + c_7 x + c_8 y + c_9 z + c_{10}$, the Hermite interpolation algorithm produces five linear equations for containment and another five for tangency:

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

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

2.1.2 *Higher-Order Interpolation for Implicit Data.* In the Hermite interpolation, smoothness is achieved by forcing normals of tangent planes of a surface to be parallel to those of given points and space curves. For some applications of geometric modeling, for example, design of ship hulls, however, more than tangent-plane continuity is desirable. The concept of smoothness is generalized by defining *a higher-order geometric continuity.* DeRose [1985] 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 order k agree along C. Warren [1986] 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 *rescaling 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 $af - bg$ up to order 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 an 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 [1991] also prove that this notion of rescaling C^k continuity is equivalent to other order k derivative continuity measures as well as to reparameterization continuity for parametric surfaces.

In Bajaj and Ihm [1992a], it is shown that, given a surface degree, the Hermite interpolation algorithm finds *all* surfaces meeting each other with C^1 continuity. Though we are 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 of a fixed degree, we can generate an interpolation matrix $\mathbf{M}_I$ whose nullspace captures an interesting proper subset of the whole class. This technique is based on the following theorem.

THEOREM 2.1. *Let $g(x, y, z)$ and $h(x, y, z)$ be distinct, irreducible polynomials. If the surfaces $g(x, y, z) = 0$ and $h(x, y, z) = 0$ intersect transversally in a single irreducible curve C, then any algebraic surface $f(x, y, z) = 0$ that meets $g(x, y, z) = 0$ with C^k rescaling continuity along C must be the form $f(x, y, z) = \alpha(x, y, z)g(x, y, z) + \beta(x, y, z)h^{k+1}(x, y, z)$. If $g(x, y, z) = 0$ and $h(x, y, z) = 0$ share no common components at infinity, then the degree* *of* $\alpha(x, y, z)g(x, y, z) \leq$ *degree of* $f(x, y, z)$ *and the degree of* $\beta(x, y, z)h^{k+1}(x, y, z) \leq$ *degree of* $f(x, y, z)$.

PROOF. If $g = 0$ and $h = 0$ intersect transversally in a single irreducible curve, then Kunz [1985, Ch. VI] and Warren [1986, Ch. III] show that the space of all surfaces interpolating $(g = 0) \cap (h = 0)$ consists of exactly those surfaces that can be expressed as $cg + gh = 0$ where c and d are polynomials in x, y, and z. Algebraically, this condition corresponds to the fact that the polynomials g and h generate a prime ideal (see Warren [1986] for more details). Macaulay [1916, p. 52] shows that this algebraic condition implies that any polynomial p, all of whose partial derivatives up to order k vanish along $(g = 0) \cap (h = 0)$, must be expressible in the form $p = \sum_{i=0}^{k+1} \gamma_i g^i h^{k+1-i}$ where the γ_i are polynomials in x, y, and z.

By the definition of C^k rescaling continuity, there must exist $a(x, y, z)$ and $b(x, y, z)$ (not identically zero on $(g = 0) \cap (h = 0)$) such that all partial derivatives of $af - bg$ up to order k vanish on $(g = 0) \cap (h = 0)$. Thus, $af - bg = \sum_{i=0}^{k+1} \beta_i g^i h^{k+1-i}$. This expression can be rewritten as $af = eg + fh^{k+1}$. Finally, since $a(x, y, z)$ is nonzero on $(g = 0) \cap (h^{k+1-i} = 0)$, it is possible to show [Warren 1986, p. 15] that f by itself can be expressed in the form $\alpha g + \beta h^{k+1}$.

The second portion of the theorem follows directly from the fact that $g = 0$ and $h = 0$ do not have any common components at infinity. A proof of this fact appears in Warren [1986, Ch. V]. □

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 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_i, and β_i. When the hypothesis of Theorem 2.1 is met, the polynomials α_i and β_i are of bounded degrees. From the relations in (1), we see that these unknown coefficients form a system of linear equations, yielding an interpolation matrix $\mathbf{M_I}$.

Example 2.2 (*Algebraic Surfaces 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 $f_2(x, y, z) = x = 0$. We compute a cubic surface $f_3(x, y, z) = 0$ which meets f_1 along C with C^2 continuity as follows. A general cubic algebraic surface is given by $f_3(x, y, z) = c_1x^3 + c_2y^3 + c_3z^3 + c_4x^2y + c_5xy^2 + c_6x^2z + c_7xz^2 + c_8y^2z + c_9yz^2 + c_{10}xyz + c_{11}x^2 + c_{12}y^2 + c_{13}z^2 + c_{14}xy + c_{15}yz + c_{16}xz + c_{17}x + c_{18}y + c_{19}z + c_{20} = 0$. Equating the generic f_3 for C^2 continuity as explained, we have $f_3(x, y, z) = (r_1x + r_2y + r_3z + r_4)f_1(x, y, z) + r_5f_2(x, y, z)^3$, yielding the linear equations: $c_1 - r_1 - r_5 = 0$, $c_2 - 2r_2 = 0$, $c_3 - 2r_3 = 0$, $c_4 - r_2 = 0$, $c_5 - 2r_1 = 0$, $c_6 - r_3 = 0$, $c_7 - 2r_1 = 0$, $c_8 - 2r_3 = 0$, $c_9 - 2r_2 = 0$, $c_{10} = 0$, $c_{11} - r_4 = 0$, $c_{12} - 2r_4 = 0$,

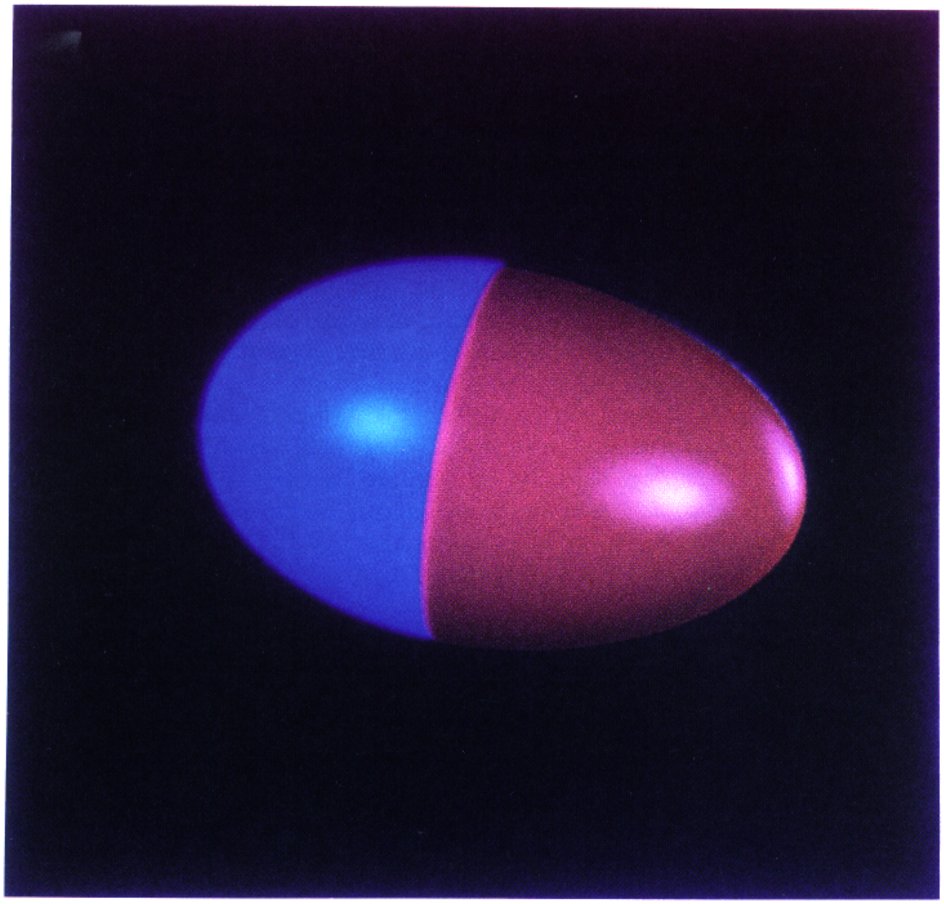

Fig. 1. A C^2 continuous algebraic surface.

$c_{13} - 2r_4 = 0$, $c_{14} = c_{15} = c_{16} = 0$, $c_{17} + 2r_1 = 0$, $c_{18} + 2r_2 = 0$, $c_{19} + 2r_3 = 0$, $c_{20} + 2r_4 = 0$ in the unknowns $c_1, \ldots, c_{20}$ and $r_1, \ldots, r_5$. By eliminating $r_1, \ldots, r_5$ from the equations, we get a homogeneous linear system $\mathbf{M_I x} = \mathbf{0}$ in terms of f_3's coefficients $c_1, \ldots, c_{20}$. An instance cubic surface $f_3(x, y, z) = -2 * y * z^2 + 2 * x * z^2 + 10 * z^2 - 2 * y^3 + 2 * x * y^2 + 10 * y^2 - x^2 * y + 2 * y + 5 * x^2 - 2 * x - 10$ is shown in Figure 1.

In the same way, we can compute quartic surfaces $f_4(x, y, z) = 16z^4 - 6yz^3 + 32xz^3 + 32z^3 + 16y^2z^2 - 16xyz^2 - 16yz^2 + 24x^2z^2 + 32xz^2 - 16y^3z + 32xy^2z + 32y^2z - 8x^2yz + 16yz + 32x^3z + 16x^2z - 32xz - 32z - 9y^4 - 16xy^3 - 16y^3 + 16x^2y^2 + 32xy^2 + 16y^2 - 8x^3y - 8x^2y + 16xy + 16y + 24x^4 + 32x^3 - 8x^2 - 32x - 16$ which meets f_3 with C^3 continuity along the curve defined by f_3 and $f_5(x, y, z) = y = 0$. See also Figure 2, where four algebraic surfaces meet four ellipses in each of the two different configurations, all with C^3 continuity.

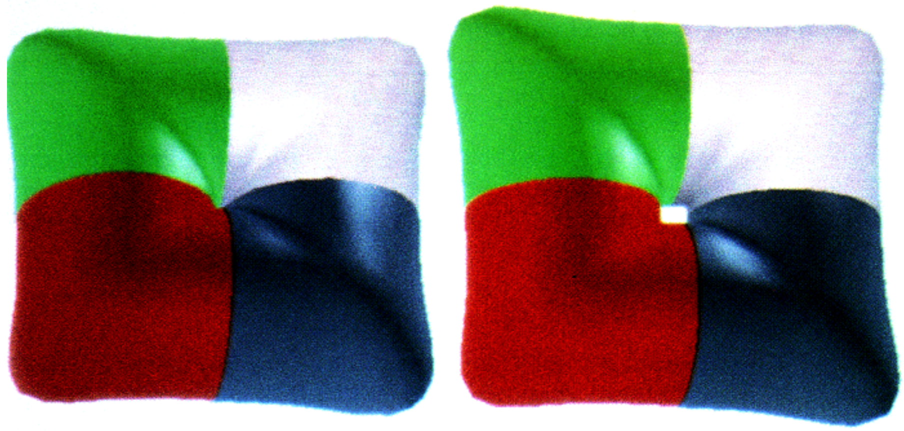

Fig. 2. C^3 continuous algebraic surfaces.

2.1.3 *Higher-Order Interpolation for Parametric Data.* There exists extensive literature on parametric curves and surfaces and their use in computer-aided geometric design; see DeRose [1985] for several references. Parametric surfaces have been successfully used in creating three-dimensional objects, and it is quite natural to have higher-order geometric information available in the parametric form. That is, geometric input information could be given in terms of a parametric surface in x, y, z space, $H(s, t) = (x(s, t), y(s, t), z(s, t))$, where $x(s, t)$, $y(s, t)$, and $z(s, t)$ are rational polynomials in s and t, and a parametric curve described by $C(u) = (s(u), t(u))$ in the s, t parameter space.

An algebraic surface $f(x, y, z) = 0$ that meets $H(s, t)$ along $H(C(u))$ with C^k continuity can again be selected from the nullspace of a properly computed matrix. As discussed in Garrity and Warren [1991], $f(x, y, z) = 0$ meets $H(s, t)$ at a smooth point $p_0 = H(C(u_0))$ with C^k continuity if and only if $f(H(s, t))$ and all the partial derivatives of $f(H(s, t))$ up to order k are zero at $C(u_0)$. Knowing this we can easily see that an algebraic surface ($f(x, y, z) = 0$ meets a parametric surface $H(s, t)$ along the entire curve $H(C)u))$ if and only if $f(H(C(u)))$ and all the partial derivatives of $f(H(C(u)))$ up to order k are zero for all values of u.

Example 2.3 (*Computation of C^2 Continuous Blobs*). Consider two circles on two spheres $H_1(s, t) = (x = (1 + 2s + s^2 + t^2)/(1 + s^2 + t^2), y = (2t)/(1 + s^2 + t^2), z = (1 - s^2 - t^2)/(1 + s^2 + t^2))$ and $H_2(s, t) = (x = (-1 + 2s - s^2 - t^2)/(1 + s^2 + t^2), y = (2t)/(1 + s^2 + t^2), z = (1 - s^2 - t^2)/(1 + s^2 + t^2))$, that are defined by $C_1(u) = C_2(u) = (s = 0, t = u)$ in the parameter space. A generic quartic algebraic surface $f(x, y, z) = 0$ has 35 unknown coefficients, and making $f(H_i(s, t))$ and all of its partial derivatives up to order two vanish along $C_i(u)$ produces 88 homogeneous linear equations in terms of the 35 unknowns. The rank of this homogeneous linear system turns out to be 32, and the surface contains three free parameters in its coefficients: $f(x, y, z) = r_3 + (-4r_1 - r_2 - 2r_3)z^2 + (5r_1 + r_2 + r_3)z^4 + (-4r_1 - r_2 - 2r_3)y^2 + (10r_1 + 2r_2 + 2r_3)y^2z^2 + (5r_1 + r_2 + r_3)y^4 + r_2x^2 + (-2r_1 - r_2)x^2z^2 + (-2r_1 - r_2)x^2y^2 + r_1x^4$. When $r_1 = -r_3 = 1$, one can use r_2 as a control coefficient for gradually changing the shapes of the surfaces that join two half spheres with C^2 continuity. Three instances of blobs are illustrated in Figure 3, for $r_1 = -r_3 = 1$, and with $r_2 = 0, -3, -5$, respectively.

2.2 Normalization

To compute an algebraic surface that least-squares approximates given data, one needs to first define a distance metric which is meaningful and computationally efficient. The *geometric* distance of a point p from a surface S: $f(x, y, z) = 0$ is the Euclidean distance from p to the nearest point on S. However, computing the geometric distance from a point to an algebraic surface itself entails an expensive computational procedure, and when it is adopted for surface approximation, the problem becomes even more intractable. A commonly used approximation to geometric distance from a point to implicitly represented algebraic curves and surfaces is the value $f(p)$, that is, the *algebraic* distance. Since $cf(x, y, z) = 0$ is the same surface for all $c \neq 0$, the coefficients of f are first normalized such that $f(x, y, z) = 0$ is a representation of the equivalence class $\{cf(x, y, z) = 0 | c \neq 0\}$.

The normalization we shall use is a *quadratic normalization* of the form $\mathbf{x}^T\mathbf{x} = 1$. While some variations [Bookstein 1979; Pratt 1987; Sampson 1982] of a quadratic normalization have been proposed in fitting scattered planar data with conic curves, it is not easily seen how different quadratic or nonquadratic normalizations affect surface fitting when the degree of a surface is greater than two, a case of considerable interest for geometric modeling. The normalization $\mathbf{x}^T\mathbf{x} = 1$ is a sphere in the coefficient vector space and does not have *singularities*. That is, this normalization only eliminates the degenerate surface with all zero coefficients as a possible solution. This normalization also leads to compact and efficient algorithms for surface fitting. It remains open to determine a generalized quadratic normalization of the form $\mathbf{x}^T\mathbf{M_N}\mathbf{x} = 1$, where $\mathbf{M_N}$ is no longer the identity matrix, with good properties for surface fitting.

2.3 Least-Squares Approximation

2.3.1 *Approximation on Algebraic Distances.* When the rank r of the interpolation matrix $\mathbf{M_I} \in \mathbf{R}^{n_i \times n_v}$ is less than the dimension n_v of the coefficient vector, there exists a family of algebraic surfaces which satisfy the given geometric constraints where the underdetermined coefficients can be homogeneously expressed in terms of q $(= n_v - r)$ free parameters. An important problem is to interactively and intuitively select a surface which is most appropriate for a given application. Selecting an instance surface from the family is equivalent to assigning values to each of the q parameters. One

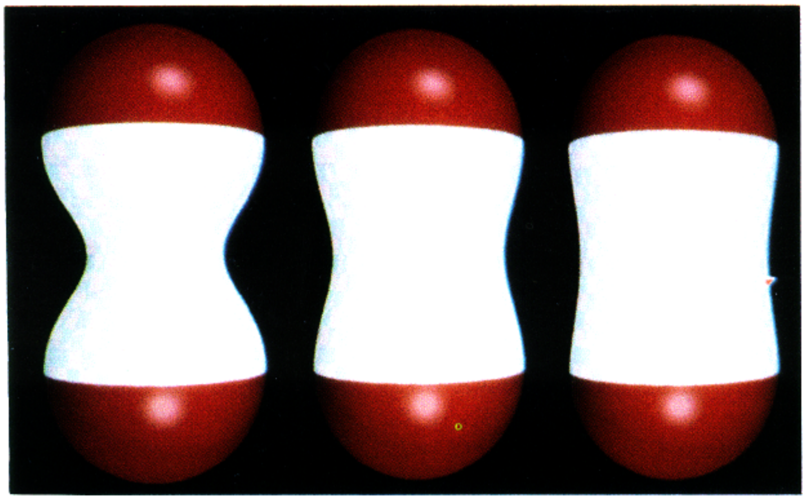

Fig. 3. A C^2 continuous algebraic surface family of blobs.

scheme for this is proposed, in Bajaj and Ihm [1992a], where interactive shape control is achieved by adjusting weights on a tetrahedral control net in the barycentric coordinate system.

Least-squares approximation can also help control the shape of the resulting algebraic surface. When there are some degrees of freedom left, we may additionally specify a set of points, curves, or surfaces around the given input data, which approximately describes a desirable surface. The final fitting surface can be obtained by consuming the remaining degrees of freedom via least-squares approximation to the additional data set.

The algebraic distance $f(p)$ is straightforward to compute and, in the case where the data point is close to a surface, approximates the geometric distance quite well. When the sum of squares of the algebraic distances of all points is minimized, we obtain algebraically nice solutions. Each row of the approximation matrix $\mathbf{M_A}$ is computed by evaluating each term in $f(x, y, z)$ at the corresponding point. And, the sum of squares is expressed as $\|\mathbf{M_A x}\|^2 = \mathbf{x}^T \mathbf{M_A^T M_A x}$.

2.3.2 *Approximation on Contour Levels.* An algebraic surface $f(x, y, z) = 0$ can be viewed as the zero contour of an explicit function $w = f(x, y, z)$. Sometimes it is more effective to generate approximation data for some level values w_i, not only $w = 0$. In Bajaj and Ihm [1992b], this contour-level approximation is found to remove unfavorable phenomena, such as singular points, self intersections, holes, etc. In this case, the objective function of our minimization problem becomes $\|\mathbf{M_A x} - \mathbf{b}\|^2$, and its computational details with examples are discussed in Bajaj and Ihm [1992b].

2.3.3 *Approximation on the First-Order Approximations of Geometric Distances.* In addition to the algebraic distance and the contour level, we look at a nonalgebraic distance metric $f(p)/\nabla f(p)$. Sampson [1982] proposed its use, in conic curve fitting, as a distance measure which is, in fact, a first-order approximation to the geometric distance. With this metric, a better approximation to the geometric distance is achievable, however, only at the expense of several iterative applications of least-squares approximation. We give an example of application of this metric to quadratic surface fitting in Section 3.

2.3.4 *Approximation on Hybrid Geometric Data.* Containment of points, curves, and surfaces is not the only way to produce $\mathbf{M_A}$. The matrix for higher-order interpolation can be used as an approximation matrix when a surface is not flexible enough for the higher-order interpolation. For example, in Example 2.2, suppose that there are more points that must be contained in the fitting surface. Then, it may not be possible that a cubic surface $f_3(x, y, z) = 0$ not only meets f_1 with C^2 rescaling continuity but also contains the extra points. If C^1 continuity is permissible, we can generate $\mathbf{M_I}$ for containment of the intersection curve (C^1), and the points (C^0) using the Hermite interpolation technique and the matrix produced in the example can be used as $\mathbf{M_A}$. That is, the remaining degrees of freedom, after interpolation, are used so that the C^2 continuity requirement is satisfied as much as possible. See Bajaj [1992] for more details.

3. COMPUTING OPTIMUM SOLUTIONS

In the foregoing section, we showed how the algebraic surface-fitting problem is transformed into a constrained quadratic minimization problem of the form:

$$\begin{aligned} &\textit{minimize} && \mathbf{x}^T \mathbf{M_A}^T \mathbf{M_A} \mathbf{x} \\ &\textit{subject to} && \mathbf{M_I} \mathbf{x} = \mathbf{0} \\ & && \mathbf{x}^T \mathbf{x} = 1, \end{aligned}$$

where $\mathbf{M_A} \in \mathbf{R}^{n_a \times n_v}$, $\mathbf{M_I} \in \mathbf{R}^{n_i \times n_v}$, and $\mathbf{x} \in \mathbf{R}^{n_v}$. This minimization problem appears in some applications [Golub and Underwood 1970]. In Golub [1973], a solution is obtained by applying Householder transformations to $\mathbf{M_I}$ to obtain its orthogonal decomposition and then by directly computing eigenvalues and eigenvectors of a reduced matrix. In this section, we consider some cases of the surface-fitting problems which arise in geometric design, and we describe different algorithms where the singular-value decomposition (SVD) is applied to computation of eigenvalues and eigenvectors. In each case, we assume a quadratic normalization constraint which always guarantees a nontrivial solution.

3.1 Interpolation Only

The first case we consider is that of interpolation alone. For example, a surface which smoothly joins four pipes is found by interpolating four curves with prescribed normals. Here, the algebraic translation of this geometric problem is *find* $\mathbf{x}$ *such that* $\mathbf{M_I x} = \mathbf{0}$ *and* $\mathbf{x}^T\mathbf{x} = 1$.

Finding a surface satisfying the given geometric specifications is equivalent to computing a nontrivial $\mathbf{x}$ in the nullspace of $\mathbf{M_I}$. In order for a nontrivial solution to exist, the degree of a surface must be high enough to guarantee that the rank r of $\mathbf{M_I}$ is less than the number n_v of coefficients. To find the nullspace in a computationally stable manner, we compute the SVD of $\mathbf{M_I}$ [Golub and Van Loan 1983] where $\mathbf{M_I}$ is decomposed as $\mathbf{M_I} = U\Sigma V^T$ where $U \in \mathbf{R}^{n_i \times n_i}$ and $V \in \mathbf{R}^{n_v \times n_v}$ are orthonormal matrices, and $\Sigma = diag(\sigma_1, \sigma_2, \ldots, \sigma_s) \in \mathbf{R}^{n_i \times n_v}$ is a diagonal matrix with diagonal elements $\sigma_1 \geq \sigma_2 \geq \cdots \geq \sigma_s \geq 0$ ($s = min\{n_i, n_v\}$). The rank r of $\mathbf{M_I}$ is the number of the positive diagonal elements of Σ, and that the last $n_v - r$ columns of V span the nullspace of $\mathbf{M_I}$. So, the nullspace of $\mathbf{M_I}$ is compactly expressed as:

$$\left\{\mathbf{x} \in \mathbf{R}^{n_v} | \mathbf{x} = \textstyle\sum_{i=1}^{n_v - r} w_i \mathbf{v}_{r+i}, \textit{ where } w_i \in \mathbf{R}, \textit{ and } \mathbf{v}_j \textit{ is the jth column of } V\right\},$$

or $\mathbf{x} = V_{n_v - r}\mathbf{w}$ where $V_{n_v - r} \in \mathbf{R}^{n_v \times (n_v - r)}$ is made of the last 4) $n_v - r$ columns of V, and $\mathbf{w}$ a $(n_v - r)$-vector.

For the quadratic normalization, we have $\mathbf{x}^T\mathbf{x} = \mathbf{x}^T V_{n_v - r}^T V_{n_v - r}\mathbf{w} = \mathbf{w}^T\mathbf{w} = 1$. Hence, any nonzero $\mathbf{x}$ from the subspace of dimension $n_v - r$ spanned by the last $n_v - r$ columns of V can be the coefficients of a surface resulting from the interpolation specified by $\mathbf{M_I}$.

Example 3.1 (*Computation of the Nullspace*). Consider $\mathbf{M_I}$ in Example 2.1. Its singular values Σ are $diag(5.65685, 4.89898, 4.89898, 2.82843, 2.82843, 2.82843, 2.0, 1.41421, 0.0, 0.0)$. Hence, the rank of $\mathbf{M_I}$ is 8, and the nullspace of $\mathbf{M_I}$ is given as $\mathbf{x} = r_1 \cdot \mathbf{v}_9 + r_2 \cdot \mathbf{v}_{10}$, where $\mathbf{v}_9 = (0.0, 0.0, 1.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0)^T$, and $\mathbf{v}_{10} = (0.57735, 0.57735, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, -0.57735)^T$. That is, the interpolating surface is $f(x, y, z) = 0.57735 r_2 x^2 + 0.57735 r_2 y^2 + r_1 z^2 - 0.57735 r_2 = 0$ which has one degree of freedom in controlling its coefficients. The nontrivial solutions are obtained by making sure that the free parameters, r_1 and r_2, do not vanish simultaneously.

Example 3.2 (*A Quintic Surface for Blending a Corner of a Table*). Generating blends that provide smooth transitions between the sharp corners and edges of solids has been recognized as one of most important problems in geometric modeling [Woodwark 1987]. A direct application of the interpolation technique to an algebraic surface is a simple way of computing a blending surface. Consider a corner of three faces that consist of the first quadrants of xy, yz, zx planes. (See Figure 4.) First, the three edges are smoothed out by three quadratic surfaces: a cone $4z^2 + 4xz - 12z + 4y^2 + 4xy - 12y + x^2 - 6x + 9 = 0$ (red checker), another cone $10yz - 25z^2 + 40z - y^2 + 10xy - 8y - 25x^2 + 40x - 16 = 0$ (blue checker), and a circular cylinder $(x-1)^2 + (y-1)^2 - 1 = 0$ (green checker). These three surfaces, intersected with $x = 0$, $y = 0$, and $z = 0$, respectively, produces

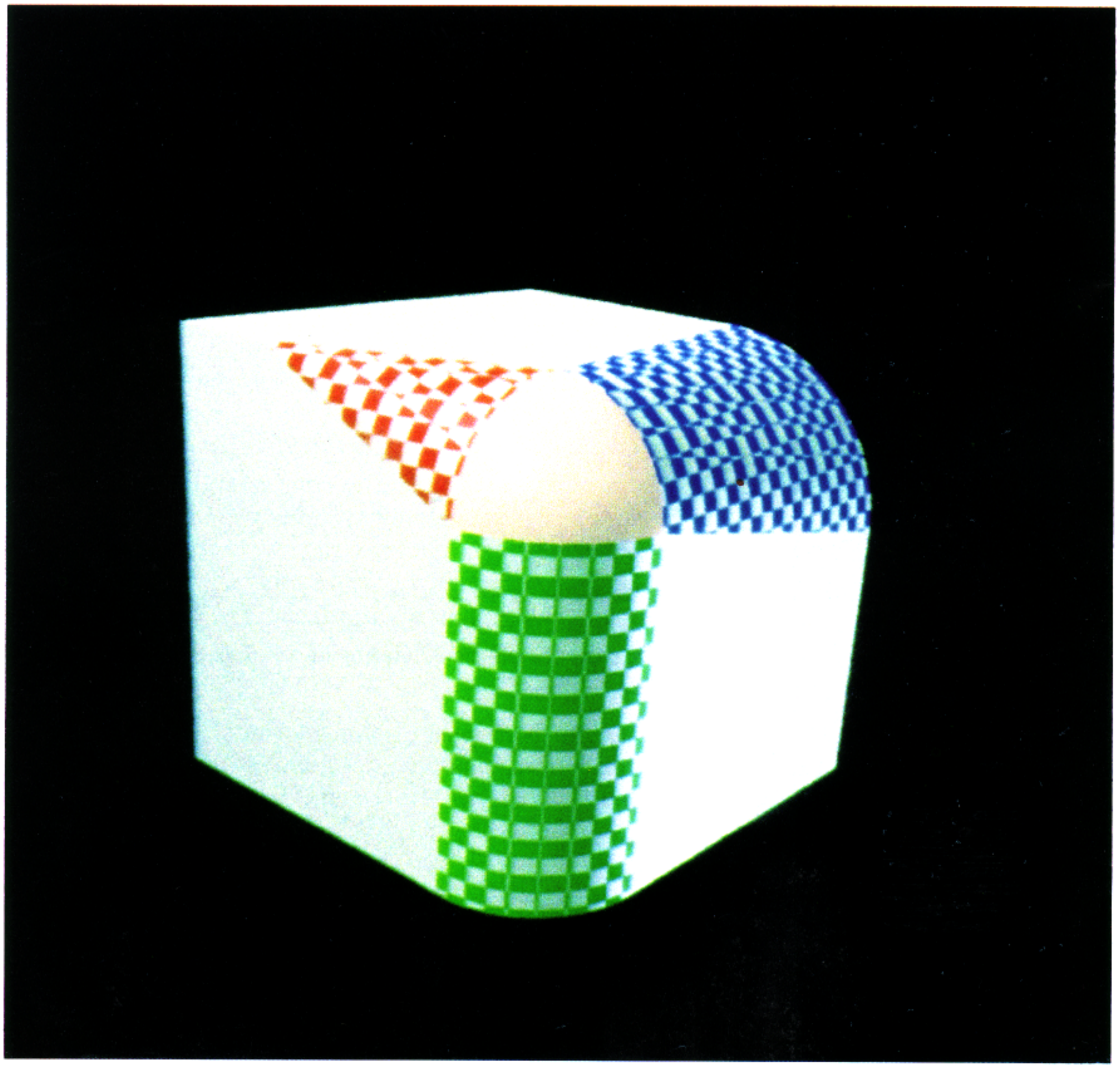

Fig. 4. Corner blending with algebraic surfaces.

three circles and associated normal vectors:

$$CIR_1: \left\{\left(1, \frac{1 + 2t + t^2}{1 + t^2}, \frac{2}{1 + t^2}\right), \left(\frac{2t + 2}{t^2 + 1}, \frac{4t}{t^2 + 1}, \frac{-2t^2 + 2}{t^2 + 1}\right)\right\},$$

$$CIR_2: \left\{\left(\frac{1 + 2t + t^2}{1 + t^2}, 1, \frac{2}{1 + t^2}\right), \left(\frac{-10t}{t^2 + 1}, \frac{2t + 2}{t^2 + 1}, \frac{5t^2 - 5}{t^2 + 1}\right)\right\},$$

$$CIR_3: \left\{\left(\frac{1 + 2t + t^2}{1 + t^2}, 21 + t^2, 1\right), \left(\frac{2t}{1 + t^2}, \frac{1 - t^2}{1 + t^2}, 0\right)\right\}.$$

Now, we look for a quintic surface S: $f(x, y, z) = 0$ which smoothly interpolates the three curves. Our Hermite interpolation technique generates a homogeneous linear system of 66 equations and 56 unknown coefficients. The rank of this 66×56 system is 45; hence, the dimension of the solution space is 11. An instance surface $f(x, y, z) = 0.0388z^5 - 0.0928yz^4 + 0.1610xz^4 - 0.2979z^4 + 0.0358y^2z^3 + 0.2155xyz^3 + 0.1259yz^3 - 0.0924x^2z^3 - 0.5050xz^3 + 0.6738z^3 - 0.0274y^3z^2 - 0.3272xy^2z^2 + 0.2301y^2z^2 - 0.1092x^2yz^2 + 0.0442xyz^2 - 0.3133yz^2 + 0.2005x^3z^2 - 0.2869x^2z^2 + 0.7299xz^2 - 0.4393z^2 - 0.0030y^4z + 0.3321xy^3z - 0.2238y^3z - 0.1341x^2y^2z + 0.0963xy^2z - 0.0541y^2z + 0.3321x^3yz - 0.4679x^2yz - 0.0340xyz + 0.4404yz - 0.1311x^4z - 0.0386x^3z + 0.4673x^2z - 0.3843xz - 0.1974z + 0.0655y^5 - 0.2883xy^4 - 0.0719y^4 + 0.0297x^2y^3 + 0.6958xy^3 - 0.1392y^3 - 0.2488x^3y^2 + 0.7195x^2y^2 - 1.4771xy^2 + 0.7541y^2 - 0.0357x^4y + 0.2427x^3y - 0.7298x^2y + 1.3307xy - 0.9488y + 0.0395x^5 - 0.0663x^4 - 0.0286x^3 + 0.3023x^2 - 0.5897x + 0.4443$ in this solution space is shown in Figure 4 (skin).

While computing the above quintic triangular patch for corner blending, it is observed that the surface is singular at the three vertices where three circles are tangent to each other even though we arrange that the normal vectors along the circles are compatible at the vertices. In Ihm [1991], it is shown that just enforcing two curves to have the same normal vector at an intersection point does not guarantee the regularity of an interpolating surface at the point. In fact, the normal vectors along the curves must be compatible up to order two at the point, and an algebraic surface *conveniently* solves this problem through singularity when the normal vectors are not compatible. We also tried a quartic algebraic surface for this blending. The rank of the 54×35 matrix computed during C^1 interpolation is 33; hence, there is a family of quartic algebraic surfaces with two degrees of freedom. However, singularity occurs along the whole circles from the cones *as well as* from the three vertices as predicted by the theory. This phenomenon is interpreted as a quartic algebraic surface is not powerful enough for these geometric data, and the algebraic degree *must be*, at least, five.

3.2 Interpolation and Approximation

In the previous subsection, $\mathbf{w}$ is a $(n_v - r)$-vector whose elements are free parameters that appear in coefficients of a family of algebraic surfaces. A final surface is selected by providing proper values for $\mathbf{w}$, by a process termed shape control. One method is for a user to specify an approximate shape of a desired surface with an additional set of points or curves and then let the system automatically find a solution vector $\mathbf{x}$. Then, what the system needs to solve efficiently is a constrained quadratic optimization problem: *minimize* $\mathbf{x}^T\mathbf{M}_\mathbf{A}^T\mathbf{M}_\mathbf{A}\mathbf{x}$ *subject to* $\mathbf{M}_\mathbf{I}\mathbf{x} = \mathbf{0}$ *and* $\mathbf{x}^T\mathbf{x} = 1$.

The solution to this minimization problem can be expressed analytically in closed form. From the interpolation constraint, we get $\mathbf{x} = V_{n_v - r}\mathbf{w}$ as before. Hence, after removing the linear constraints, we are led to the problem *minimize* $\mathbf{w}^T V_{n_v - r}^T \mathbf{M}_\mathbf{A}^T\mathbf{M}_\mathbf{A} V_{n_v - r}\mathbf{w}$ *subject to* $\mathbf{w}^T\mathbf{w} = 1$. Note that $V_{n_v - r}^T \mathbf{M}_\mathbf{A}^T\mathbf{M}_\mathbf{A} V_{n_v - r}$ is a positive definite matrix, and this problem is equivalent to minimizing the ratio of two quadratics $R(\mathbf{w}) = (\mathbf{w}^T V_{n_v-r}^T \mathbf{M}_\mathbf{A}^T \mathbf{M}_\mathbf{A} V_{n_v-r} \mathbf{w})/(\mathbf{w}^T \mathbf{w})$. $R(\mathbf{w})$, which is called *Rayleigh's quotient*, is minimized by the first eigenvector $\mathbf{w} = \mathbf{w}_{min}$ of $V_{n_v-r}^T \mathbf{M}_\mathbf{A}^T \mathbf{M}_\mathbf{A} V_{n_v-r}$, and its minimum value is the smallest eigenvalue λ_{min} [Strang 1988].

Contrary to computing the eigenvectors and eigenvalues of $V_{n_v-r}^T \mathbf{M}_\mathbf{A}^T \mathbf{M}_\mathbf{A} V_{n_v-r}$ directly as in Golub [1973], we apply singular-value decomposition to $\mathbf{M}_\mathbf{A} V_{n_v-r}$ without computing $V_{n_v-r}^T \mathbf{M}_\mathbf{A}^T \mathbf{M}_\mathbf{A} V_{n_v-r}$ explicitly. This leads to a numerically cheaper computation. Here, we assume that $n_a \geq n_v - r$, and that the rank of $\mathbf{M}_\mathbf{A} V_{n_v-r}$ is $n_v - r$. (That is, there must be enough linear constraints to consume the remaining degrees of freedom.) Then, $\mathbf{M}_\mathbf{A} V_{n_v-r} = P \Omega Q^T$ where $P \in \mathbf{R}^{n_a \times n_a}$ and $Q \in \mathbf{R}^{(n_v-r) \times (n_v-r)}$ are orthonormal matrices, and $\Omega = diag(\omega_1, \omega_2, \ldots, \omega_{n_v-r}) \in \mathbf{R}^{n_a \times (n_v-r)}$ with $\omega_1 \geq \omega_2 \geq \cdots \geq \omega_{n_v-r} > 0$.

Now,

$$\begin{aligned}
\lambda \mathbf{w} &= V_{n_v-r}^T \mathbf{M}_\mathbf{A}^T \mathbf{M}_\mathbf{A} V_{n_v-r} \mathbf{w} \\
&= Q \Omega^T P^T P \Omega Q^T \mathbf{w} \\
&= Q \Omega^T \Omega Q^T \mathbf{w}.
\end{aligned}$$

Here, $\Omega^T \Omega$ is a $(n_v - r) \times (n_v - r)$ diagonal matrix with a diagonal entry $\omega_i^2 > 0$, $i = 1, 2, \ldots, (n_v - r)$. Then, from the above equation, $\Omega^T \Omega (Q^T \mathbf{w}) = \lambda (Q^T \mathbf{w})$ which implies that the first eigenvector $\mathbf{w}_{min}$ of $V_{n_v-r}^T \mathbf{M}_\mathbf{A}^T \mathbf{M}_\mathbf{A} V_{n_v-r}$ is such that $Q^T \mathbf{w}_{min} = e_{n_v-r}$ where $e_{n_v-r} = [0, 0, \ldots, 0, 1]^T$ is a $(n_v - r)$-vector, and its minimum value λ_{min} is $\omega_{n_v-r}^2$. Hence, $\mathbf{w}_{min}$ is the last column of Q. Once we compute Q, we get the coefficients of the algebraic surface $\mathbf{x} = X_{n_v-r} Q e_{n_v-r}$, which is nontrivial, and hence satisfies the normalization constraint.

Example 3.3 (*Two Quartic Surfaces for Smoothly Joining Four Cylindrical Surfaces*). In this example, we compute a surface S: $f(x, y, z) = 0$ which smoothly joins four cylinders which are given as $CY\,L_1$: $y^2 + z^2 - 1 = 0$ for $x \geq 2$, $CY\,L_2$: $y^2 + z^2 - 1 = 0$ for $x \leq -2$, $CY\,L_3$: $x^2 + y^2 - 1 = 0$ for $z \geq 2$, and $CY\,L_4$: $x^2 + y^2 - 1 = 0$ for $z \leq -2$.

The interpolation requirement is for S to meet the four curves on the cylinders with C^1 continuity. Interpolation for a quartic surface S generates $\mathbf{M}_\mathbf{I} \in \mathbf{R}^{64 \times 35}$ (64 linear equations and 35 coefficients) whose rank is 33.[2] This implies a 2-parameter family of quartic surfaces satisfying the interpolation constraints.

Then we need to select a surface with desired shape from this family. We use least-squares approximation during this process. To illustrate the effect of approximation, two sets of points are chosen: $S_1 = \{(0, 1.75, 0), (0, -1.75, 0), (-1, 1.25, 0), (-1, -1.25, 0), (1, 1.25, 0), (1, -1.25, 0)\}$ 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.)

For the least-squares approximation with normalization, the eigenvalues and eigenvectors for S_1 and S_2 are computed. As a result, we obtain

[2]As a byproduct of interpolation process, it is found out that degree 4 is the required minimum.

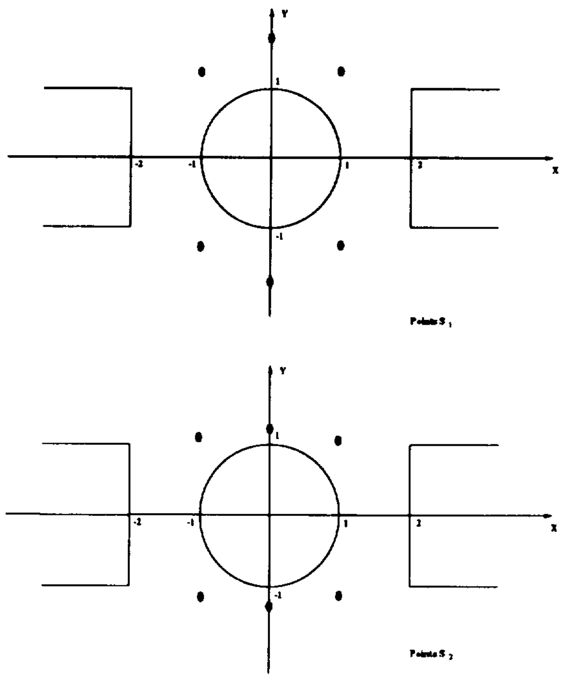

Fig. 5. Points to be approximated.

$\lambda_{min_{S_1}} = 1.265429 \cdot 10^{-1}$, $\lambda_{min_{S_2}} = 5.097809 \cdot 10^{-3}$, and $f_{S_1}(x, y, z) = 0.315034x^2 + 0.273947y^2 + 0.315034z^2 - 0.849216 - 0.035612x^4 - 0.030137x^2y^2 - 0.030137x^2z^2 + 0.005474y^4 - 0.030137y^2z^2 - 0.035612z^4$, $f_{S_2}(x, y, z) = 0.281104x^2 + 0.615461y^2 + 0.281104z^2 - 0.201225 + 0.005325x^4 - 0.323706x^2y^2 - 0.323706x^2z^2 - 0.329031y^4 - 0.323706y^2z^2 + 0.005325z^4$. The surfaces are shown in Figure 6.

Example 3.4 (*Smoothing a Solid with Triangular Algebraic Surface Patches*). Generation of a mesh of C^1 surface patches that smooths a solid has been an important problem in computer-aided geometric design. Bajaj and Ihm [1992b] present an efficient algorithm for generating a family of curved solid objects with boundary topology related to an input solid. Given a solid with triangular faces, each edge is replaced by a conic or a cubic curve, depending on the geometry around the edge, which is, then, associated with normal vectors of the same degree along the curved edge. Each face, now made of three curves and associated normals, is fleshed with a degree five, six, or seven algebraic surface patch using the interpolation and approximation technique. In this local scheme, the shape of the resulting curved solid is controlled mainly by controlling shapes of boundary curves. Figure 7 illustrates a solid, made of 100 triangular faces, with curved edges (a) and a smoothed solid (b).

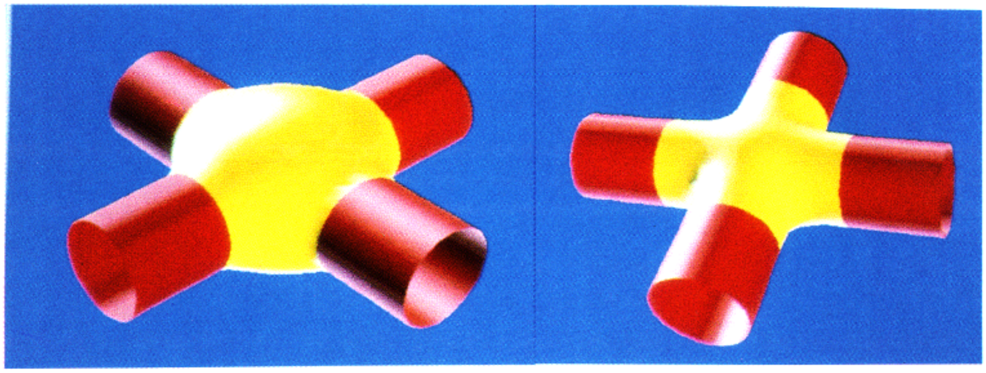

Fig. 6. Two different least-squares approximations.

3.3 Least-Squares Approximation Only

At times, we desire a surface which is only a least-squares approximation from given geometric data. This is often the case when straightforward interpolation leads to a prohibitively high algebraic degree of the resulting surface. This least-squares problem, by itself, is just a special case ($\mathbf{M}_\mathbf{I} = \mathbf{0}$) of the minimization problem in the previous section. In this case, $V_{n_v - r}$ disappears in the solution, which results in $\mathbf{x} = Qe_{n_v}$.

Example 3.5 (*Least-Squares Approximation to Points*: *Algebraic Distance*). Consider that we are computing a quadric surface $f(x, y, z) = 0$ which approximates the following collection of points in the least-squares sense: $S = \{(0.451663, -0.623974, 0.068396), (0.328349, -0.677433, -0.058929), (0.439221, -0.591023, -0.112337), (0.203666, -0.713408, -0.179608), (0.316158, -0.642981, -0.235268), (0.416159, -0.548033, -0.283786), (-0.013524, -0.726372, -0.357700), (0.091099, -0.689252, -0.414074), (0.086853, -0.728675, -0.279685), (0.189591, -0.627654, -0.465215), (0.198347, -0.674498, -0.335774), (0.350256, -0.445695, -0.535944), (0.385507, -0.498320, -0.425321), (0.277729, -0.544938, -0.506400), (0.299997, -0.596124, -0.382735)\}$.

Each row of $\mathbf{M}_\mathbf{A}$ is obtained by simply evaluating, at each point, the basis of quadrics: $\{x^2, y^2, z^2, xy, yz, zx, x, y, z, 1\}$. After applying SVD to $\mathbf{M}_\mathbf{A}$, we get a quadric surface whose error-of-fit is $\lambda_{min} = 2.281646 \cdot 10^{-7}$.

In the previous example, what is minimized is the sum of squares of the *algebraic* distance, which is the contour level of the function $w = f(x, y, z)$. This algebraic distance is not always the same as the *geometric* distance, which is the actual distance from a point to a surface. Sometimes, it is more desirable to minimize the sum of squares of the *real* distance. Unfortunately, this nonalgebraic metric entails an intractable minimization problem whose solution cannot be expressed analytically in closed form. Sampson

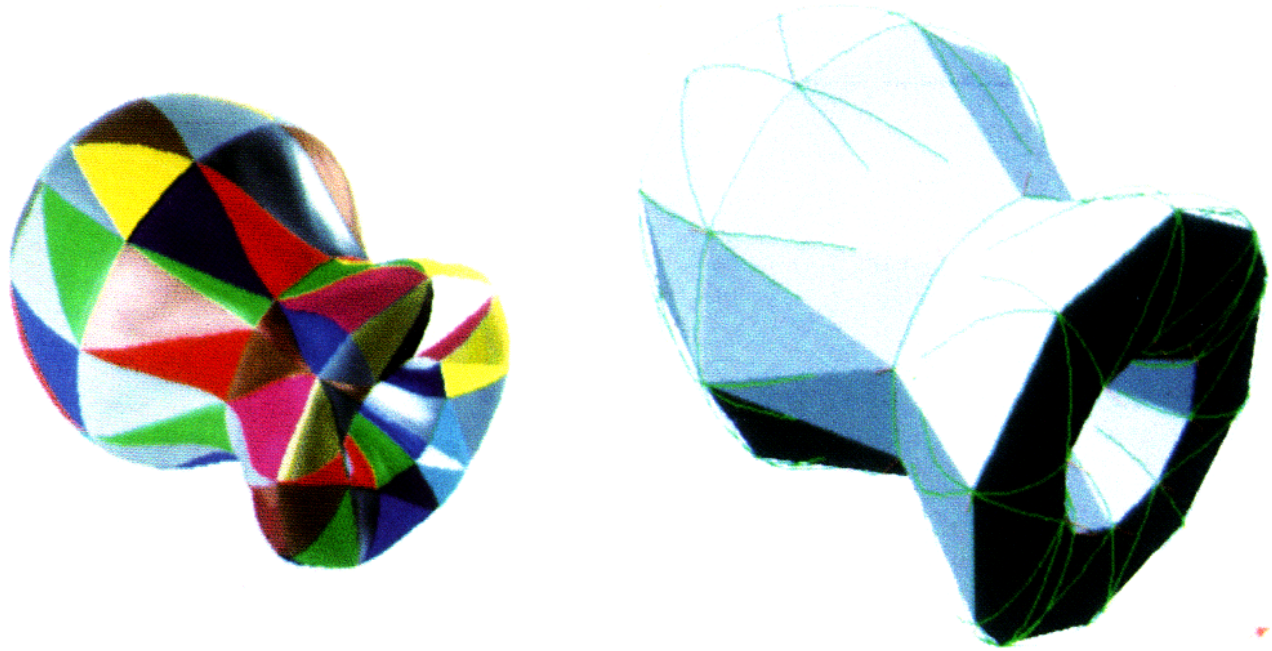

Fig. 7. Smoothing a solid with algebraic surface patches.

[1982] uses a nonalgebraic distance metric, which approximates geometric distance, in fitting conic curves. This concept can be naturally extended to the surface-fitting problem. We get to this nonalgebraic metric via a different derivation.

First, let us recall that the distance from a point p to a surface $f(x, y, z) = 0$ is the distance from p to a nearest point on the surface. Let q be the point on the surface which results in the distance. Then, the line in the direction of the normal of f at q must pass through p, and $q = p + t(\nabla f(q))/\|\nabla f(q)\|$ where the absolute value of t is the distance. From Taylor's expansion,

$$0 = f(q) = f(p) + \nabla f(p) \cdot \left(t \frac{\nabla f(q)}{\|\nabla f(q)\|} \right) + \cdots.$$

Hence,

$$|t| \approx \left| \frac{-f(p)\|\nabla f(q)\|}{\nabla f(p) \cdot \nabla f(q)} \right| \tag{2}$$

is the first-order approximation to the distance from p to f. When p is close to the surface, $\nabla f(p)$ is a good approximation to $\nabla f(q)$. In this case, the expression (2) becomes

$$\begin{aligned} |t| &\approx \left| \frac{-f(p)\|\nabla f(p)\|}{\nabla f(p) \cdot \nabla f(p)} \right| \\ &= \left| \frac{-f(p)\|\nabla f(p)\|}{\|\nabla f(p)\|^2} \right| \\ &= \frac{|f(p)|}{\|\nabla f(p)\|} \stackrel{\text{def}}{=} dist_f(p). \end{aligned}$$

This argument suggests that $dist_f(p)$, the weighted algebraic distance, be a good approximation to the geometric distance, and that

$$\sum_{for\ all\ p} dist_f(p)^2 = \sum_{for\ all\ p} \frac{f(p)^2}{\|\nabla f(p)\|^2} \tag{3}$$

be minimized instead of

$$\sum_{for\ all\ p} f(p)^2. \tag{4}$$

However, the solution which minimizes the expression (3) cannot be easily expressed in closed form due to introduction of the weight $\|\nabla f(p)\|$.

This numerical intractability can be avoided by an iterative refinement algorithm. First, we compute $\mathbf{x}_{(0)}$, coefficients of a surface $f_{(0)}$, such that (4), the sum of squares of algebraic distances, is minimized. To do this, $\mathbf{M_A} = \mathbf{M_{A(0)}}$ is obtained as before. The gradient of $f_{(0)}$ gives an initial guess to $\nabla f(p)$. Then, dividing each row of $\mathbf{M_A}$ by $\|\nabla f_{(0)}(p)\|$ for each corresponding p results in $\mathbf{M_{A(1)}}$ which is, then, singular-value-decomposed to compute $\mathbf{x}_{(1)}$ and $f_{(1)}$.

Table 1. The Geometric and Algebraic Distances

k	*geo. distance*	*alg. distance*
0	3.925480319e-05	2.281646641e-07
1	2.870799913e-05	2.497249375e-07
2	2.762911566e-05	2.472207775e-07
3	2.696617975e-05	2.465526346e-07
4	2.661304527e-05	2.461413816e-07
5	2.642308921e-05	2.459224774e-07
6	2.632187346e-05	2.458047987e-07
7	2.626807583e-05	2.457421127e-07
8	2.623953195e-05	2.457087993e-07
9	2.622440016e-05	2.456911254e-07
...	...	...
26	2.620735209e-05	2.456712015e-07
27	2.620735193e-05	2.456712014e-07
28	2.620735184e-05	2.456712013e-07

This process is repeated further, producing a sequence of $f_{(k)}$ which refines the solution. In each iteration, $f_{(k)}$ is expected to be a better approximation to the surface we are trying to find.

Example 3.6 (*Iterative Weighted Least-Squares Approximation to Points: Nonalgebraic Distance*). In Example 3.5, we have computed $\mathbf{M_A} = \mathbf{M_{A(0)}}$, and $f_{(0)}$. Table I illustrates the result of application of the iterative algorithm to the set of points. The *geo. distance* column shows the sum of squares of the *real* geometric distance[3] for $f_{(k)}$, and the *alg. distance* column shows the value of the expression (4), the sum of squares of the algebraic distances for $f_{(k)}$. It is observed that the sum of squares of the geometric distances decreases as iterations proceed, which implies that $f_{(k)}$ converges to the surface which best fits the given point data. It is also interesting to see that the sum of squares of the algebraic distances makes a quantum jump at the first iteration and then converges to a *local minimum*.

4. CONCLUDING REMARKS

We have implemented our algebraic surface-fitting algorithms and included them in SHASTRA, a collaborative distributed geometric design and manipulation environment [Anupam and Bajaj 1993]. SHASTRA consists of a growing set of X-11 toolkits for geometric design, that are networked into a highly extensible environment where all the toolkits are interoperable.

[3]The geometric distance was calculated by solving a 4-by-4 system of nonlinear equations, derived using the Lagrange multiplier method.

In this article, we have seen that implicitly represented algebraic surfaces can be very appropriate for C^k interpolation and approximation to geometric data. However, there still remain some difficult problems to be solved for algebraic surfaces to be fully used for geometric modeling. First, it is not always easy to make sure that input points and curves lie on one real component of the solution surface. One heuristic, which can be used, is to include auxiliary points and curves to effectively bridge the gap between separate surface components. Another approach is proposed in Moore and Warren [1991] where a *signed-distance* fit is used to generate a single sheet of a surface inside a tetrahedron for densely scattered point data. However, the question remains open for producing necessary and sufficient conditions on the coefficients of the fitting surfaces, which would ensure that all given points and curves lie on the same continuous real-surface component.

Another unfavorable issue in algebraic surface design is that of singularities. We need to be able to ensure that singularities do not occur on some interior region of a surface that we are interested in. For example, the triangular algebraic surface patch in Figure 4 is singular at the three vertices while it is regular inside. In fact, singularities are useful in this case because no regular patch can smoothly blend the corner for that particular configuration. It is highly desirable again to derive polynomial constraints required to guarantee that any specified algebraic surface patch is without any singularities.

ACKNOWLEDGMENT

We wish to thank Professor Robert Lynch for his valuable comments on the matrix computations and Dr. Guoliang Xu for several discussions on geometric continuity. We are also indebted to Vinod Anupam, Susan Evans, Andrew Royappa, and Guoliang Xu for their help in the implementations of the algorithms in SHASTRA.

REFERENCES

ALBANO A. 1974. Representation of digitized contours in terms of conic arcs and straight line segments. *Comput. Graph. Image Process. 3*, 23–33.

ANUPAM, V., AND BAJAJ, C. 1993. Collaborative multimedia scientific design in SHASTRA. In *Proceedings of the 1993 ACM Multimedia Conference*. ACM Press, New York, 447–456.

BAJAJ, C. 1993. The emergence of algebraic curves and surfaces in geometric design. In *Directions in Geometric Computing*. Information Geometers Press, United Kingdom, 1–27.

BAJAJ, C. 1992. Surface fitting with implicit algebraic surface patches. In *Topics in Surface Modeling*. SIAM Publications, Philadelphia, Pa., 23–52.

BAJAJ, C. 1988. Geometric modeling with algebraic surfaces. In *The Mathematics of Surfaces III*. Oxford University Press, New York, 3–48.

BAJAJ, C., AND IHM, I. 1992a. Algebraic surface design with Hermite interpolation. *ACM Trans. Graph. 11*, 1 (Jan.), 61–91.

BAJAJ, C., AND IHM, I. 1992b. Smoothing polyhedra using implicit algebraic splines. *Comput. Graph. 26*, 2 (July) 79–88.

BAJAJ, C., AND XU, G. 1992. A Splines: Local interpolation and approximation using C^k-continuous piecewise real algebraic curves. Computer Science Tech. Rep. CAPO-92-95, Purdue Univ., West Lafayette, Ind.

BAJAJ, C., HOFFMANN, C., HOPCROFT, J., AND LYNCH, R. 1988. Tracing surface intersections. *Comput. Aided Geom. Des. 5*, 285–307.

BLOOMENTHAL, J. 1988. Polygonization of implicit surfaces. *Comput. Aided Geom. Des. 5*, 341–355.

BOOKSTEIN, F. 1979. Fitting conic sections to scattered data. *Comput. Graph. Image Process. 9*, 56–71.

DEROSE, A. 1985. Geometric continuity: A parametrization independent measure of continuity for computer aided geometric design. PhD thesis, Computer Science, Univ. of California, Berkeley.

FAROUKI, R. 1988. The approximation of nondegenerate offset surfaces. *Comput. Aided Geom. Des. 3*, 15–43.

GARRITY, T., AND WARREN, J. 1991. Geometric continuity. *Comput. Aided Geom. Des. 8*, 51–65.

GNANADESIKAN, R. 1977. *Methods for Statistical Data Analysis of Multivariate Observations*. John Wiley & Sons, New York.

GOLUB, G. 1973. Some modified matrix eigenvalue problems. *SIAM Rev. 15*, 2, 318–334.

GOLUB, G., AND UNDERWOOD, R. 1970. Stationary values of the ratio of quadratic forms subject to linear constraints. *Z. Agnew. Math. Phys. 21*, 318–326.

GOLUB, G., AND VAN LOAN, C. 1983. *Matrix Computation*. The Johns Hopkins University Press, Baltimore, Md.

HOFFMANN, C., AND HOPCROFT, J. 1987. The potential method for blending surfaces and corners. In *Geometric Modeling: Algorithms and New Trends*. SIAM, Philadelphia, Pa., 347–366.

IHM, I. 1991. On surface design with implicit algebraic surfaces. PhD thesis, Purdue Univ., West Lafayette, Ind.

KUNZ, E. 1985. *Introduction to Commutative Algebra and Algebraic Geometry*. Birkhauser, New York.

MACAULAY, F. 1916. *The Algebraic Theory of Modular Systems*. Cambridge University Press, London, U.K.

MOORE, D., AND WARREN, J. 1991. Approximation of dense scattered data using algebraic surfaces. In *Proceedings of the 24th Hawaii International Conference on System Sciences* (Kauai, Hawaii). IEEE Computer Society Press, Washington, D.C., 681–690.

OWEN, J., AND ROCKWOOD, A. 1987. Blending surfaces in solid modeling. In *Geometric Modeling: Algorithms and New Trends*. SIAM, Philadelphia, Pa., 367–383.

PRATT, V. 1987. Direct least squares fitting of algebraic surfaces. *Comput. Graph. 21*, 3, 145–152.

SAMPSON, P. 1982. Fitting conic sections to very scattered data: An iterative refinement of the bookstein algorithm. *Comput. Graph. Image Process. 18*, 97–108.

SEDERBERG, T. 1990a. Techniques for cubic algebraic surfaces. *IEEE Comput. Graph. Appl. 10*, 4 (July), 14–25.

SEDERBERG, T. 1990b. Techniques for cubic algebraic surfaces. *IEEE Comput. Graph. Appl. 10*, 5 (Sept.), 12–21.

SEDERBERG, T. 1985. Piecewise algebraic surface patches. *Comput. Aided Geom. Des. 2*, 1–3, 53–59.

STRANG, G. *Linear Algebra and Its Applications*. 3rd ed. Harcourt Brace Jovanovich, San Diego, Calif.

WARREN, J. 1989. Blending algebraic surfaces. *ACM Trans. Graph. 8*, 4, 263–278.

WARREN, J. 1986. On algebraic surfaces meeting with geometric continuity. PhD thesis, Cornell Univ., Ithaca, N.Y.

WOODWARK, J. 1987. Blends in geometric modelling. In *The Mathematics of Surfaces*, Volume 2. Oxford University Press, New York, 255–297.

Received May 1990; revised May 1991; accepted February 1992