

Using Multivariate Resultants to Find the Intersection of Three Quadric Surfaces

ENG-WEE CHIONH
National University of Singapore

RONALD N. GOLDMAN
Rice University
and

JAMES R. MILLER
University of Kansas

Macaulay's concise but explicit expression for multivariate resultants has many potential applications in computer-aided geometric design. Here we describe its use in solid modeling for finding the intersections of three implicit quadric surfaces. By Bézout's theorem, three quadric surfaces have either at most eight or infinitely many intersections. Our method finds the intersections, when there are finitely many, by generating a polynomial of degree at most eight whose roots are the intersection coordinates along an appropriate axis. Only addition, subtraction, and multiplication are required to find the polynomial. But when there are possibilities of extraneous roots, division and greatest common divisor computations are necessary to identify and remove them.

Categories and Subject Descriptors: I.3.5 [Computer Graphics]: Computational Geometry and Object Modeling—*curve, surface and object representations; geometric algorithms, languages and systems; physically based modeling*

General Terms: Algorithm, Design, Theory

1. INTRODUCTION

Quadric surfaces appear with surprising regularity on the boundary of physical objects commonly modeled in computer-aided design and manufacturing (CAD/CAM) systems. However, only bounded portions of these surfaces actually lie on the boundary of the model. The challenge for computer-aided

This work was supported by the National University of Singapore, by the National Science Foundation (NSF) under grant DMC-8813688, and by the University of Kansas under General Research allocation 3760-X0-0038.

Authors' addresses: E. W. Chionh, Department of Information Systems and Computer Science, National University of Singapore, Singapore; R. N. Goldman, Department of Computer Science, Rice University, Houston, TX; J. R. Miller, Department of Computer Science, University of Kansas, 415 Snow Hall, Lawrence, KS 66045-2192.

Permission to copy without fee all or part of this material is granted provided that the copies are not made or distributed for direct commercial advantage, the ACM copyright notice and the title of the publication and its date appear, and notice is given that copying is by permission of the Association for Computing Machinery. To copy otherwise, or to republish, requires a fee and/or specific permission.

© 1991 ACM 0730-0301/91/1000-0366 \$01.50

ACM Transactions on Graphics, Vol. 10, No. 4, October 1991, Pages 378–400.

geometric design (CAGD) is to determine those portions of the surfaces that do lie on the object and to represent them in the database.

Relevant portions of surfaces are generally determined from Boolean operations on quadric half-spaces. As a result, the pieces of the surfaces that lie on the model are bounded by curves of intersection between pairs of quadric surfaces. In general these curves are irreducible space quartics, although in certain situations they are reducible, either to a line plus a space cubic or to a pair of (possibly degenerate) conics. Like the surfaces from which they arise, it is typically the case that only portions of these curves actually lie on the boundary of the object. The endpoints of the curves are the locations in space where the complete curve intersects yet a third quadric surface.

Computing the intersection curves and partitioning them at their intersections with other surfaces is one of the primary tasks of the Boundary Evaluation algorithm in solid modeling systems. This algorithm generates the boundary representation of the solid resulting from a Boolean operation applied to two other solids. Boundary evaluation is of vital importance to solid modeling systems since so many application functions and interaction techniques require a complete and accurate representation of the bounding faces, edges, and vertices of a solid. It is also generally the most complex piece of software in such systems, owing to the many complex geometric relationships that can arise which must be handled properly in an automatic fashion.

The Boundary Evaluation algorithm operating on solids A and B can be summarized at a high level as follows [19]:

1. Generate a sufficient set of tentative edges.
 - (a) Intersect each surface on the boundary of A with each surface on the boundary of B.
 - (b) Partition the original edges of A, the original edges of B, and the new curves generated in part (a) at their intersections with other surfaces.
2. Classify each partitioned edge as INSIDE, OUTSIDE, or ON (the boundary of) the resulting solid.
3. Retain only those edges whose classification is ON.

Robust algorithms which address step 1(a) in the context of the so-called natural quadrics have been described in Miller [13], O'Connor [17], and Piegel [18]. General algebraic techniques not restricted to the natural quadrics are presented in Levin [8]. Sarraga [21] has also studied these techniques in the context of the natural quadrics. Ocken et al. [16] have reported general algebraic intersection schemes applicable to any pair of rational surfaces (of which quadric surfaces are a subset). More recently, Farouki et al. [6] have described general algebraic methods for automatically intersecting any pair of quadrics, detecting degenerate results such as conics and other rational parametric curve branches in the process.

We focus in this paper on issues related to part (b) of step 1. At least 3 methods of partitioning curves are known. Since none of the methods perform satisfactorily (or are even applicable) in all situations, a given system will typically implement more than one technique and then use which ever one is best for a particular geometric configuration. Briefly, the known methods are the following (see Miller [12] for further details and references).

- (1) *Curve-surface intersection.* Directly intersect the quadric surface intersection curve with other quadric surfaces on the model. This approach is commonly used when the curve is a straight line or a conic section.
- (2) *Curve-curve intersection.* If an intersection curve between surfaces S1 and S2 intersects (nontangentially) another surface S3, then it must be the case that both S1 and S2 intersect S3 along distinct intersection curves. The partitions could then be determined by intersecting any 2 of these 3 curves. Again this technique is most commonly employed when the two chosen curves are conic sections; however, Levin describes a rather elaborate algebraic scheme for intersecting two general nonplanar quartic intersection curves [8].
- (3) *Three-quadric intersection.* It is clear from the discussion to this point that the bounding vertices of the curves lie at points common to three quadric surfaces. Partitioning the intersection curves by pursuing this observation is most commonly used as a fall-back approach when no two of the surfaces intersect in straight lines or conic sections. A scheme to find such points based on the application of quick heuristics with general, but slower, fall-back numerical methods is described in Morgan and Sarraga [14]. The remainder of this paper describes a new approach using multivariate resultants to find the points in common to 3 quadric surfaces.¹

Using multivariate resultants [22] to find the intersections of 3 implicit quadric surfaces is a new approach in CAGD.² It provides new insight to the intersection problem by isolating the algebraic details into a single resultant expression. At one stroke a polynomial of degree at most 8 is found, whose roots are the intersection coordinates along one of the coordinate axes. When the polynomial is a nonzero constant (degree zero), there is no intersection in affine space; when it is identically zero, the intersection is infinite (a curve). We refer to such polynomials derived from resultants as *intersection polynomials*.

Our method differs from those using the more commonly known Sylvester resultants [20]. With Sylvester resultants, an intersection polynomial can be found by successive elimination: first compute 3 resultants of degrees at most 4 by eliminating one variable; then compute 3 more resultants of degrees at most 16 by eliminating another variable; finally compute the greatest common divisor (GCD) of the latter 3 resultants. This method has several drawbacks. Computationally it is very expensive; 6 resultants and the GCD of 3 polynomials of degrees up to 16 must be computed. The degrees of the intersection polynomials produced may exceed 8, and more seriously, it is not clear how to detect and remove extraneous roots—roots which are introduced

¹While the curve and surface intersection techniques surveyed above, as well as the specific implementation of the 3 quadric intersection algorithms to be described, are applicable only to quadric surfaces, they are nonetheless useful in modeling systems supporting higher order surfaces, possibly in addition to quadrics. We expand upon two aspects of this observation in Section 6.

²This use of multivariate resultants has also been discussed by Bajaj et al. [1].

during the derivation of the intersection polynomial, but which do not represent the coordinates of any intersections. In contrast, multivariate resultants produce, in a single step, an intersection polynomial of degree at most 8. This approach also allows the identification of extraneous roots when they exist.

In the following sections we briefly introduce resultants and describe Macaulay's method³ for finding them. The use of multivariate resultants for computing the intersections of 3 implicit quadric surfaces is presented, and special techniques for practical implementation are discussed. These techniques include the avoidance of division in Macaulay's multivariate resultant expression, systematic detection of degree deficiencies, and removal of extraneous roots. An evaluation of the accuracy and robustness of this method is also reported.

2. MACAULAY'S MULTIVARIATE RESULTANT EXPRESSION

Resultants are an important tool in elimination theory. They are polynomial functions of the coefficients of a system of homogeneous (all terms having the same degree) polynomial equations, whose vanishing is a necessary and sufficient condition for the system of homogeneous equations to have a nontrivial (not all zero) common solution. For k homogeneous polynomial equations in k variables, the resultant always exists. Bézout and Sylvester resultants are well-known resultant expressions for $k = 2$ (see Salmon, [20]). For $k \geq 2$, a concise but explicit resultant expression was provided by Macaulay [10]. We use the term "multivariate resultants" for the case $k > 2$, since in this case a resultant eliminates more than two variables from a system of homogeneous polynomial equations. This differs from the terminology of Collins [5] whose multivariate resultants refer to the case $k = 2$ but with indeterminate coefficients.

2.1 The Algorithm

Macaulay's multivariate resultant expression for k homogeneous polynomial equations in k variables is a quotient of two determinants $|D|/|M|$ where the denominator $|M|$ is a factor of the numerator $|D|$ and M itself is a submatrix of D . D and M can be constructed by the following algorithm adapted from Macaulay [10]. Important facts are given as comments in the algorithm. Their proofs can be found in Macaulay [10]. When the degrees of the k homogeneous polynomials are all equal, a similar method by Macaulay produces smaller matrices D and M [4, 9].

Input. k homogeneous polynomials f_1, \dots, f_k of degrees n_1, \dots, n_k in x_1, \dots, x_k .
 $t \leftarrow 1 + \sum_1^k (n_i - 1)$
 $\Omega' \leftarrow \{x_1^{i_1} \cdots x_k^{i_k} \mid i_1 + \cdots + i_k = t\}$

³In his ACM 1987 Doctoral Dissertation Award-winning thesis, Canny [2] discussed and applied multivariate resultants, using a formulation originally due to Hurwitz. Subsequently he also discovered expressions for multivariate resultants due to Cayley and Macaulay. We discovered Macaulay resultants independently in May 1988 while reading his *The Algebraic Theory of Modular Systems* [11].

Comment. Ω^t is the set of all degree t homogeneous monomials in x_1, \dots, x_k .

$T \leftarrow \Omega^t$

for i from 1 to k do

$S \leftarrow$ set of monomials $\in T$ divisible by x_i^n

$\Omega_{i-1} \leftarrow \frac{S}{x_i^n}$

$T \leftarrow T - S$

Comment. Ω^t is partitioned as $x_1^n \Omega_0 \cup \dots \cup x_k^n \Omega_{k-1}$.

for i from 1 to k do

 for each monomial $\omega^{(t-n)}$ in Ω_{i-1} , compute the polynomial $\omega^{(t-n)} f_i$

Comment. $\binom{t+k-1}{k-1}$ homogeneous polynomials of degree t are formed.

$D \leftarrow$ coefficient matrix of these polynomials with Ω^t as column indices

for i from 0 to $k-2$ do

$F_i \leftarrow$ set of powers $\in \Omega_i$ divisible by at least one of x_{i+2}^n, \dots, x_k^n

$M \leftarrow$ the minor of D with

 column indices $x_1^n F_0 \cup \dots \cup x_{k-1}^n F_{k-2}$ and rows from polynomials

$f_1 F_0, \dots, f_{k-1} F_{k-2}$

Output. $|D| / |M|$.

2.2 An Example

To illustrate the algorithm, consider three homogeneous polynomials a , b , and c :

$$\begin{aligned} a &\equiv a_1 x^2 + a_2 xy + a_3 xz + a_4 y^2 + a_5 yz + a_6 z^2 = 0 \\ b &\equiv b_1 x^2 + b_2 xy + b_3 xz + b_4 y^2 + b_5 yz + b_6 z^2 = 0 \\ c &\equiv c_1 x + c_2 y + c_3 z = 0. \end{aligned} \tag{1}$$

By writing the monomial $x^i y^j z^k$ as ijk , we have

$$\Omega^3 = \{300, 210, 201, 120, 111, 102, 030, 021, 012, 003\}$$

$$\Omega_0, \Omega_1 = \{100, 010, 001\}$$

$$\Omega_2 = \{110, 101, 011, 002\}$$

$$F_0, F_1 = \{001\}$$

and

$$\frac{|D|}{|M|} = \begin{vmatrix} a_1 & a_2 & a_3 & a_4 & a_5 & a_6 & \cdot & \cdot & \cdot & \cdot \\ \cdot & a_1 & \cdot & a_2 & a_3 & \cdot & a_4 & a_5 & a_6 & \cdot \\ \cdot & \cdot & \mathbf{a}_1 & \cdot & a_2 & a_3 & \cdot & \mathbf{a}_4 & a_5 & a_6 \\ b_1 & b_2 & b_3 & b_4 & b_5 & b_6 & \cdot & \cdot & \cdot & \cdot \\ \cdot & b_1 & \cdot & b_2 & b_3 & \cdot & b_4 & b_5 & b_6 & \cdot \\ \cdot & \cdot & \mathbf{b}_1 & \cdot & b_2 & b_3 & \cdot & \mathbf{b}_4 & b_5 & b_6 \\ \cdot & \cdot & c_1 & \cdot & c_2 & c_3 & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & c_2 & c_3 & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & c_1 & \cdot & \cdot & c_2 & c_3 & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot & c_1 & \cdot & \cdot & c_2 & c_3 \end{vmatrix} \div \begin{vmatrix} a_1 & a_4 \\ b_1 & b_4 \end{vmatrix} \tag{2}$$

where the dots in (2) represent zero entries. Note that M is the submatrix of D with column indices $(2\ 0\ 0)$ F_0 , $(0\ 2\ 0)$ F_1 and rows from $(0\ 0\ 1)$ a and $(0\ 0\ 1)$ b .

The construction of D is greatly facilitated by arranging the monomials in the *lexicographic* order. For example, powers 120, 300, and 210 are arranged in the order 300, 210, 120.

2.3 Equivalent Resultant Expressions

If we arrange (1) as c, a, b and the variables as y, x, z for input to the algorithm, the resultant expression will be

$$\frac{|D|}{|M|} = \begin{vmatrix} c_1 & c_2 & c_3 & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & c_1 & \cdot & c_2 & c_3 & \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & c_1 & \cdot & c_2 & c_3 & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & c_1 & \cdot & \cdot & c_2 & c_3 & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & c_1 & \cdot & \cdot & c_2 & c_3 & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot & c_1 & \cdot & \cdot & c_2 & c_3 \\ a_1 & a_2 & a_3 & a_4 & a_5 & a_6 & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & a_1 & \cdot & a_2 & a_3 & \cdot & a_4 & a_5 & a_6 \\ b_1 & b_2 & b_3 & b_4 & b_5 & b_6 & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & b_1 & \cdot & b_2 & b_3 & \cdot & b_4 & b_5 & b_6 \end{vmatrix} \div \begin{vmatrix} c_2 & \cdot \\ \cdot & c_2 \end{vmatrix}. \quad (3)$$

It can be verified by direct computation using Maple [3] that (2) and (3) are equivalent, except possibly for sign. In fact, there are $(k!)^2$ ways of expressing Macaulay resultants for a system of k homogeneous polynomials in k variables. To indicate that a resultant expression is generated by the permutations σ and δ on the equations and variables, the resultant expression is denoted as $(f_{\sigma_1}^{n_{\sigma_1}} f_{\sigma_2}^{n_{\sigma_2}} f_{\sigma_3}^{n_{\sigma_3}})_{x_{\delta_1} x_{\delta_2} x_{\delta_3}}$ where $f_1^{n_1}, f_2^{n_2}, f_3^{n_3}$ are homogeneous polynomials in the variables x_1, x_2, x_3 of degree n_1, n_2, n_3 , respectively. For examples, resultant expressions (2), (3) are denoted as $(a^2 b^2 c^1)_{xyz} (c^1 a^2 b^2)_{yxz}$, respectively.

3. MACAULAY RESULTANTS WITHOUT DIVISION

Since Macaulay's multivariate resultant expression is a quotient of two determinants, division is required when $k > 2$. This is undesirable because in our context the determinant entries will be polynomials and some specializations of determinant entries will result in 0/0. This problem can be avoided by expanding and dividing the determinants to arrive at an integral expression before specializing the entries. But this will usually result in hundreds and thousands of terms. Fortunately division can be avoided without losing the conciseness of the resultant expression by the following two techniques.

3.1 Resultant Expressions Conducive to Division

The first technique involves choosing the "right" Macaulay expression and using the determinant identity $|\alpha + \beta, \gamma| = |\alpha, \gamma| + |\beta, \gamma|$. For example, (3) is one of the 36 equivalent resultant expressions for the system $a^2 b^2 c^1$ in

the variables x, y, z . Expanding by cofactors of the seventh column and ignoring sign, we find that

$$(c^1 a^2 b^2)_{yxx} = \begin{vmatrix} c_1 & c_2 & c_3 & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & c_1 & \cdot & c_2 & c_3 & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & c_1 & \cdot & c_2 & c_3 & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & c_1 & \cdot & c_2 & c_3 & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot & c_1 & \cdot & c_2 & c_3 \\ a_1 & a_2 & a_3 & a_4 & a_5 & a_6 & \cdot & \cdot & \cdot \\ \cdot & \cdot & a_1 & \cdot & a_2 & a_3 & a_4 & a_5 & a_6 \\ b_1 & b_2 & b_3 & b_4 & b_5 & b_6 & \cdot & \cdot & \cdot \\ \cdot & \cdot & b_1 & \cdot & b_2 & b_3 & b_4 & b_5 & b_6 \end{vmatrix} \div c_2. \quad (4)$$

But the determinant in (4) can be written as the sum of two determinants by splitting the second column as

$$\begin{vmatrix} c_1 & c_2 & c_3 & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & c_2 & c_3 & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & c_1 & \cdot & c_2 & c_3 & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & c_1 & \cdot & c_2 & c_3 & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot & c_1 & \cdot & c_2 & c_3 \\ a_1 & \cdot & a_3 & a_4 & a_5 & a_6 & \cdot & \cdot & \cdot \\ \cdot & \cdot & a_1 & \cdot & a_2 & a_3 & a_4 & a_5 & a_6 \\ b_1 & \cdot & b_3 & b_4 & b_5 & b_6 & \cdot & \cdot & \cdot \\ \cdot & \cdot & b_1 & \cdot & b_2 & b_3 & b_4 & b_5 & b_6 \end{vmatrix} + \begin{vmatrix} c_1 & \cdot & c_3 & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & c_1 & \cdot & c_2 & c_3 & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & c_1 & \cdot & c_2 & c_3 & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & c_1 & \cdot & c_2 & c_3 & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot & c_1 & \cdot & c_2 & c_3 \\ a_1 & a_2 & a_3 & a_4 & a_5 & a_6 & \cdot & \cdot & \cdot \\ \cdot & \cdot & a_1 & \cdot & a_2 & a_3 & a_4 & a_5 & a_6 \\ b_1 & b_2 & b_3 & b_4 & b_5 & b_6 & \cdot & \cdot & \cdot \\ \cdot & \cdot & b_1 & \cdot & b_2 & b_3 & b_4 & b_5 & b_6 \end{vmatrix}. \quad (5)$$

Clearly, the first determinant is divisible by c_2 , which is a factor of the second column. The same procedure can be repeated with the second determinant. Eventually we will arrive at a determinant whose entries do not involve c_2 ; this determinant must be zero since it must be divisible by c_2 . Consequently the resultant for the system $a^2 b^2 c^1$ can be written as a linear combination of five 9×9 determinants with no division whatsoever.

3.2 Partial Specialization

The other technique may be called partial specialization—introducing zero values into the determinant expressions to facilitate division. For example,

one of the 36 resultant expressions for the system $a^2b^2c^2$ is

$$(a^2b^2c^2)_{yzx} = \frac{\begin{vmatrix} a_1 & a_2 & a_3 & a_4 & a_5 & a_6 & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & a_1 & \cdot & a_2 & a_3 & \cdot & a_4 & a_5 & a_6 & \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & a_1 & \cdot & a_2 & a_3 & \cdot & a_4 & a_5 & a_6 & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & a_1 & \cdot & \cdot & a_2 & a_3 & \cdot & \cdot & a_4 & a_5 & a_6 & \cdot \\ \cdot & \cdot & \cdot & \cdot & a_1 & \cdot & \cdot & a_2 & a_3 & \cdot & \cdot & a_4 & a_5 & a_6 \\ \cdot & \cdot & \cdot & \cdot & \cdot & a_1 & \cdot & \cdot & a_2 & a_3 & \cdot & \cdot & a_4 & a_5 & a_6 \\ b_1 & b_2 & b_3 & b_4 & b_5 & b_6 & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & b_1 & \cdot & b_2 & b_3 & \cdot & b_4 & b_5 & b_6 & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & b_1 & \cdot & b_2 & b_3 & \cdot & b_4 & b_5 & b_6 & \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & b_1 & \cdot & \cdot & b_2 & b_3 & \cdot & \cdot & b_4 & b_5 & b_6 & \cdot \end{vmatrix}}{\begin{vmatrix} \cdot & \cdot & \cdot & \cdot & \cdot & b_1 & \cdot & \cdot & b_2 & b_3 & \cdot & \cdot & b_4 & b_5 & b_6 \\ c_1 & c_2 & c_3 & c_4 & c_5 & c_6 & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & c_1 & \cdot & c_2 & c_3 & \cdot & c_4 & c_5 & c_6 & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & c_1 & \cdot & c_2 & c_3 & \cdot & c_4 & c_5 & c_6 & \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & c_1 & \cdot & \cdot & c_2 & c_3 & \cdot & \cdot & c_4 & c_5 & c_6 & \cdot \end{vmatrix}} \div \begin{vmatrix} a_4 & a_6 & \cdot \\ a_1 & a_4 & \cdot \\ b_4 & b_6 & \cdot \end{vmatrix}. \quad (6)$$

Note that the denominator reduces to $a_4^2b_6$ when $a_6 = 0$. Furthermore, if $b_4 = c_4 = 0$, then successive expansion by cofactors of the 11th, 12th, 15th, and 7th columns gives

$$(a^2b^2c^2)_{yzx} = a_4 \begin{vmatrix} a_1 & a_2 & a_3 & a_4 & a_5 & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & a_1 & \cdot & a_2 & a_3 & a_4 & a_5 & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot & a_1 & \cdot & a_2 & a_3 & a_4 & a_5 & \cdot & \cdot & \cdot \\ b_1 & b_2 & b_3 & \cdot & b_5 & b_6 & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & b_1 & \cdot & b_2 & b_3 & \cdot & b_5 & b_6 & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & b_1 & \cdot & b_2 & b_3 & \cdot & b_5 & b_6 & \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & b_1 & \cdot & b_2 & b_3 & \cdot & b_5 & b_6 & \cdot & \cdot & \cdot \\ c_1 & c_2 & c_3 & \cdot & c_5 & c_6 & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & c_1 & \cdot & c_2 & c_3 & \cdot & c_5 & c_6 & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & c_1 & \cdot & c_2 & c_3 & \cdot & c_5 & c_6 & \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & c_1 & \cdot & c_2 & c_3 & \cdot & c_5 & c_6 & \cdot & c_5 & c_6 \end{vmatrix}. \quad (7)$$

Thus when $a_6 = b_4 = c_4 = 0$, the resultant for the system $a^2b^2c^2$ is an expression involving no division.

3.3 Triangularization

To use the technique of partial specialization, the “right” zeros must be introduced into the Macaulay expressions. This can be done systematically

by triangularizing the matrix

$$\begin{pmatrix} a_4 & a_5 & a_6 \\ b_4 & b_5 & b_6 \\ c_4 & c_5 & c_6 \end{pmatrix} \quad (8)$$

using Gaussian elimination.

By interchanging rows if necessary, the structure of matrix (8) can be reduced to one of the following eight configurations, where an \times represents an entry which can be either zero or nonzero:

I			II			III			IV		
0	0	0	0	0	1	0	1	\times	0	1	\times
0	0	0	0	0	0	0	0	0	0	0	1
0	0	0	0	0	0	0	0	0	0	0	0
V			VI			VII			VIII		
1	\times	\times	1	\times	\times	1	\times	\times	1	\times	\times
0	0	0	0	0	1	0	1	\times	0	1	\times
0	0	0	0	0	0	0	0	0	0	0	1

Thus given any system $a^2b^2c^2$, a system with the same common solutions and with $a'_6 = b'_4 = c'_4 = 0$ can always be derived; so $(a^2b^2c^2)_{yzz}$ can always be written as (7) and no division is involved. But when the rank of matrix (8) is less than 3, simpler expressions without division can be obtained by similar techniques, as is shown later.

4. THE METHOD

By using multivariate resultants, we show that the intersection problem reduces to solving for the roots of a single univariate polynomial and for each root solving three simultaneous polynomial equations in two variables. Note that not all real roots of an intersection polynomial correspond to real quadric intersections for they may be paired with complex valued solutions in the other axes. Solving univariate polynomials is a well-understood numerical problem [15]; solving three simultaneous polynomial equations in two variables can be done by finding the three pairwise Sylvester resultants and then solving the GCD of the pairwise resultants. The following sections provide the details of this use of multivariate resultants.

4.1 Finding the Intersection Polynomials

The general equations of three quadric surfaces can be written as

$$\begin{aligned} x^2 \begin{bmatrix} A_1 \\ B_1 \\ C_1 \end{bmatrix} + y^2 \begin{bmatrix} A_2 \\ B_2 \\ C_2 \end{bmatrix} + z^2 \begin{bmatrix} A_3 \\ B_3 \\ C_3 \end{bmatrix} + xy \begin{bmatrix} A_4 \\ B_4 \\ C_4 \end{bmatrix} + yz \begin{bmatrix} A_5 \\ B_5 \\ C_5 \end{bmatrix} + zx \begin{bmatrix} A_6 \\ B_6 \\ C_6 \end{bmatrix} \\ + x \begin{bmatrix} A_7 \\ B_7 \\ C_7 \end{bmatrix} + y \begin{bmatrix} A_8 \\ B_8 \\ C_8 \end{bmatrix} + z \begin{bmatrix} A_9 \\ B_9 \\ C_9 \end{bmatrix} + \begin{bmatrix} A_{10} \\ B_{10} \\ C_{10} \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}. \end{aligned} \quad (10)$$

By regarding any one of the variables x , y , or z as an indeterminate constant and introducing a homogenizing variable, the quadric equations can be written as homogeneous equations like (11), where x has been taken as the indeterminate constant.

$$w_x^2 \begin{bmatrix} A_1 x^2 + A_7 x + A_{10} \\ B_1 x^2 + B_7 x + B_{10} \\ C_1 x^2 + C_7 x + C_{10} \end{bmatrix} + w_x y \begin{bmatrix} A_4 x + A_8 \\ B_4 x + B_8 \\ C_4 x + C_8 \end{bmatrix} + w_x z \begin{bmatrix} A_6 x + A_9 \\ B_6 x + B_9 \\ C_6 x + C_9 \end{bmatrix} \\ + y^2 \begin{bmatrix} A_2 \\ B_2 \\ C_2 \end{bmatrix} + yz \begin{bmatrix} A_5 \\ B_5 \\ C_5 \end{bmatrix} + z^2 \begin{bmatrix} A_3 \\ B_3 \\ C_3 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}. \quad (11)$$

We refer to (11) as the x -homogenized equations of (10). This should not be confused with the homogeneous equations of (10) which are

$$x^2 \begin{bmatrix} A_1 \\ B_1 \\ C_1 \end{bmatrix} + y^2 \begin{bmatrix} A_2 \\ B_2 \\ C_2 \end{bmatrix} + z^2 \begin{bmatrix} A_3 \\ B_3 \\ C_3 \end{bmatrix} + xy \begin{bmatrix} A_4 \\ B_4 \\ C_4 \end{bmatrix} + yz \begin{bmatrix} A_5 \\ B_5 \\ C_5 \end{bmatrix} + zx \begin{bmatrix} A_6 \\ B_6 \\ C_6 \end{bmatrix} \\ + xw \begin{bmatrix} A_7 \\ B_7 \\ C_7 \end{bmatrix} + yw \begin{bmatrix} A_8 \\ B_8 \\ C_8 \end{bmatrix} + zw \begin{bmatrix} A_9 \\ B_9 \\ C_9 \end{bmatrix} + w^2 \begin{bmatrix} A_{10} \\ B_{10} \\ C_{10} \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}. \quad (12)$$

The resultant R_x of (11) can be obtained using the Macaulay expression for the system $a^2 b^2 c^2$ with the substitutions

$$\begin{array}{lll} a_1 = A_1 x^2 + A_7 x + A_{10} & a_2 = A_4 x + A_8 & a_3 = A_6 x + A_9 \\ b_1 = B_1 x^2 + B_7 x + B_{10} & b_2 = B_4 x + B_8 & b_3 = B_6 x + B_9 \\ c_1 = C_1 x^2 + C_7 x + C_{10} & c_2 = C_4 x + C_8 & c_3 = C_6 x + C_9 \\ & a_4 = A_2 & a_5 = A_5 & a_6 = A_3 \\ & b_4 = B_2 & b_5 = B_5 & b_6 = B_3 \\ & c_4 = C_2 & c_5 = C_5 & c_6 = C_3 \end{array} \quad (13)$$

It is clear that R_x is a polynomial in x . By Bézout's theorem [22], the degree of R_x should, in general be 8. This is not obvious from the resultant expression but can be established theoretically by using the isobaric property of the resultant which we now explain. Each coefficient of (11) is assigned an integer value known as the weight. This is done by first arbitrarily choosing any one variable from w_x , y , z . The weight of a coefficient is then set equal to the power of that variable in the term where the coefficient occurs. The weight of a term in a resultant expression is defined to be the sum of the weights of the coefficients appearing in that term. With this weight assignment, it is known that resultants are isobaric (all terms have the same weight), and the weight of each term is equal to the product of degrees of the given equations [11]. Thus by choosing the variable w_x , the weights of the

coefficients of (11) are as follows: 2 for a_1, b_1, c_1 ; 1 for $a_2, a_3, b_2, b_3, c_2, c_3$; and 0 for the rest. But here the coefficients are polynomials in x , and clearly their weights are simply their degrees. Consequently the resultant is isobaric of weight $2 \times 2 \times 2$ means each term has a weight of 8. It follows that the degree of x in the resultant expression is at most 8.

As an example, the x , y , and z intersection polynomials for the quadrics (14)

$$\begin{aligned} x^2 \begin{bmatrix} 1 \\ 2 \\ 1 \end{bmatrix} + y^2 \begin{bmatrix} -2 \\ -4 \\ -2 \end{bmatrix} + z^2 \begin{bmatrix} 3 \\ 6 \\ 3 \end{bmatrix} + xy \begin{bmatrix} 1 \\ 17 \\ 1 \end{bmatrix} + yz \begin{bmatrix} -1 \\ -5 \\ 0 \end{bmatrix} + zx \begin{bmatrix} 1 \\ -2 \\ -1 \end{bmatrix} \\ + x \begin{bmatrix} -1 \\ -2 \\ -1 \end{bmatrix} + y \begin{bmatrix} 2 \\ 4 \\ 2 \end{bmatrix} + z \begin{bmatrix} 1 \\ 2 \\ 1 \end{bmatrix} + \begin{bmatrix} -25 \\ -50 \\ -25 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix} \end{aligned} \quad (14)$$

are (15), (16), and (17), respectively:

$$x^4(x-1)(19x+25)(x^2-x-25) = 0 \quad (15)$$

$$y^4(y-2)(19y+50)(2y^2-2y+25) = 0 \quad (16)$$

$$z^4(z-3)(19z+75)(3z^2+z-25) = 0. \quad (17)$$

Since the vanishing of the resultant is a necessary and sufficient condition for the solvability of a system of polynomial equations, each root x of $R_x = 0$ corresponds to a solution (w_x, y, z) of (11). If $w_x \neq 0$, then $(x, y/w_x, z/w_x)$ is a solution of (10). But it is also possible that a root x may correspond to a solution $(0, y, z)$ of (11). Such x 's do not correspond to a solution of (10). Consequently the intersection problem reduces to solving for the roots of a single univariate polynomial $R_x = 0$, provided that R_x is a nonzero nonconstant polynomial and we know what roots of $R_x = 0$ to discard.

Clearly similar arguments apply if y or z , instead of x , is considered to be an indeterminate constant.

4.2 Identically Zero Resultants

An identically zero resultant for (11) means that a nontrivial solution exists regardless of the value of x . For example, $(w_x, y, z) = (0, -\beta, \alpha)$ is always a nontrivial solution of (18) for any value of x .

$$\begin{aligned} w_x^2 \begin{bmatrix} A_1 x^2 + A_7 x + A_{10} \\ B_1 x^2 + B_7 x + B_{10} \\ C_1 x^2 + C_7 x + C_{10} \end{bmatrix} + w_x y \begin{bmatrix} A_4 x + A_8 \\ B_4 x + B_8 \\ B_4 x + B_8 \end{bmatrix} + w_x z \begin{bmatrix} A_6 x + A_9 \\ B_6 x + B_9 \\ B_6 x + B_9 \end{bmatrix} \\ + y^2 \begin{bmatrix} \alpha p_1 \\ \alpha p_2 \\ 0 \end{bmatrix} + yz \begin{bmatrix} \alpha q_1 + \beta p_1 \\ \alpha q_2 + \beta p_2 \\ 0 \end{bmatrix} + z^2 \begin{bmatrix} \beta q_1 \\ \beta q_2 \\ 0 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}. \end{aligned} \quad (18)$$

This means that the resultant R_x , a polynomial in x , has infinitely many roots and thus is identically zero. But notice that in this case (18) could have been homogenized as (19), which is a system of homogeneous equations of degrees 2 2 1. In fact, there is a correspondence between rank deficiencies of matrix (8) and degree deficiencies of the homogenized quadric equations. For ranks 2, 1, and 0, the homogenized quadric equations are of degrees 2 2 1, 2 1 1, and 1 1 1, respectively. They are given by (19), (20), and (21):

$$\begin{aligned}
& (A_1 x^2 + A_7 x + A_{10})w_x^2 + (A_4 x + A_8)w_x y \\
& \quad + (A_6 x + A_9)w_x z + A_2 y^2 + A_5 yz + A_3 z^2 = 0 \\
& (B_1 x^2 + B_7 x + B_{10})w_x^2 + (B_4 x + B_8)w_x y \\
& \quad + (B_6 x + B_9)w_x z + B_2 y^2 + B_5 yz + B_3 z^2 = 0 \\
& (C_1 x^2 + C_7 x + C_{10})w_x + (C_4 x + C_8)y + (C_6 x + C_9)z = 0
\end{aligned} \tag{19}$$

$$\begin{aligned}
& (A_1 x^2 + A_7 x + A_{10})w_x^2 + (A_4 x + A_8)w_x y \\
& \quad + (A_6 x + A_9)w_x z + A_2 y^2 + A_5 yz + A_3 z^2 = 0 \\
& (B_1 x^2 + B_7 x + B_{10})w_x + (B_4 x + B_8)y + (B_6 x + B_9)z = 0 \\
& (C_1 x^2 + C_7 x + C_{10})w_x + (C_4 x + C_8)y + (C_6 x + C_9)z = 0
\end{aligned} \tag{20}$$

$$\begin{aligned}
& (A_1 x^2 + A_7 x + A_{10})w_x + (A_4 x + A_8)y + (A_6 x + A_9)z = 0 \\
& (B_1 x^2 + B_7 x + B_{10})w_x + (B_4 x + B_8)y + (B_6 x + B_9)z = 0 \\
& (C_1 x^2 + C_7 x + C_{10})w_x + (C_4 x + C_8)y + (C_6 x + C_9)z = 0.
\end{aligned} \tag{21}$$

When the rank is 0 or 1, the degree of the intersection polynomial is at most 4 or 6, respectively. This does not contradict Bézout's theorem because x, y, z are affine coordinates but the theorem is true only in projective space.

In many cases, this provision of degree deficiencies will overcome the problem of identically zero resultants. Exceptional cases still exist like (22):

$$\begin{aligned}
& (A_1 x^2 + A_7 x + A_{10})w_x^2 + (A_4 x + A_8)w_x y + (A_6 x + A_9)w_x z \\
& \quad + \alpha p_1 y^2 + (\alpha q_1 + \beta p_1)yz + \beta q_1 z^2 = 0 \\
& (B_1 x^2 + B_7 x + B_{10})w_x^2 + (B_4 x + B_8)w_x y + (B_6 x + B_9)w_x z \\
& \quad + \alpha p_2 y^2 + (\alpha q_2 + \beta p_2)yz + \beta q_2 z^2 = 0 \\
& (C_1 x^2 + C_7 x + C_{10})w_x + \alpha(p_3 x + q_3)y + \beta(p_3 x + q_3)z = 0
\end{aligned} \tag{22}$$

where $(w_x, y, z) = (0, -\beta, \alpha)$ is again a nontrivial solution for all values of x . In this case, y or z rather than x could be taken as the indeterminate constant to produce an intersection polynomial provided the given quadric

equations do not have the following structure:

$$s^2(y^2) + r^2(z^2) = rs(yz),$$

$s(xy) - r(zx), s(y) - r(z), (y^2), (z^2), (yz)$ are coplanar

$$q^2(z^2) + p^2(x^2) = pq(zx),$$

$q(yz) - p(xy), q(z) - p(x), (z^2), (x^2), (zx)$ are coplanar

$$v^2(x^2) + u^2(y^2) = uv(xy),$$

$v(zx) - u(yz), v(x) - u(y), (x^2), (y^2), (xy)$ are coplanar

where $(x^2), (y^2), \dots$, are the coefficients of x^2, y^2, \dots , written as column vectors like those in (10). An example of quadric equations with this structure is

$$\begin{aligned} x^2 \begin{bmatrix} 1 \\ 0 \\ 0 \\ 0 \end{bmatrix} + y^2 \begin{bmatrix} 0 \\ 1 \\ 0 \\ 0 \end{bmatrix} + z^2 \begin{bmatrix} 0 \\ 0 \\ 1 \\ 0 \end{bmatrix} + xy \begin{bmatrix} \frac{v}{u} \\ u \\ \frac{v}{0} \\ 0 \end{bmatrix} + yz \begin{bmatrix} 0 \\ s \\ r \\ s \end{bmatrix} + zx \begin{bmatrix} \frac{p}{q} \\ 0 \\ q \\ p \end{bmatrix} \\ + x \begin{bmatrix} \alpha \\ q\theta_y \\ u\theta_z \end{bmatrix} + y \begin{bmatrix} r\theta_x \\ \beta \\ v\theta_z \end{bmatrix} + z \begin{bmatrix} s\theta_x \\ p\theta_y \\ \gamma \end{bmatrix} + \begin{bmatrix} k_1 \\ k_2 \\ k_3 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix} \end{aligned} \quad (23)$$

where $sqv = rpu$. For systems like (23), the resultants of the x , y , and z -homogenized quadrics are all identically zero. Transforming the given quadric equations is one way to get around this problem. For example, after applying transformation (24), the method can be used to find x' , y' , and z' . An inverse transformation will then give x , y , and z . However, we do not know if a transformation will always work, and if so what transformation is most efficient.

$$\begin{bmatrix} x' \\ y' \\ z' \end{bmatrix} = \begin{pmatrix} 1 & 1 & 1 \\ 1 & 1 & 0 \\ 1 & 0 & 0 \end{pmatrix} \begin{bmatrix} x \\ y \\ z \end{bmatrix}. \quad (24)$$

In many situations an identically zero resultant may actually indicate the presence of degeneracies which enable the solutions to be found in a simpler manner. For example, when the three equations involve only one variable, all intersections polynomials will be identically zero.

4.3 Extraneous Roots

A root x of $R_x = 0$ that corresponds to a root $(0, y, z)$ of the given quadrics does not constitute a root (x, y, z) of the given quadrics. These x roots are extraneous roots; they can be found by solving for those x for which the system of equations derived by setting w_x to 0 in (11), (19), (20), or (21) has nontrivial common solutions. For (11), there are no such x values; for (19),

(20), or (21), these x values are roots of the polynomial G_x , which is the GCD of the pairwise Sylvester resultants which are, respectively:

$$\begin{vmatrix} A_2 & A_5 & A_3 \\ B_5 & B_3 & 0 \\ 0 & B_5 & B_3 \end{vmatrix}, \quad \begin{vmatrix} B_5 & B_3 \\ C_4 x + C_8 & C_6 x + C_9 \end{vmatrix} \quad (25)$$

$$\begin{vmatrix} A_2 & A_5 & A_3 \\ B_4 x + B_8 & B_6 x + B_9 & 0 \\ 0 & B_4 x + B_8 & B_6 x + B_9 \end{vmatrix},$$

$$\begin{vmatrix} A_2 & A_5 & A_3 \\ C_4 x + C_8 & C_6 x + C_9 & 0 \\ 0 & C_4 x + C_8 & C_6 x + C_9 \end{vmatrix}, \quad \begin{vmatrix} B_4 x + B_8 & B_6 x + B_9 \\ C_4 x + C_8 & C_6 x + C_9 \end{vmatrix} \quad (26)$$

$$\begin{vmatrix} A_4 x + A_8 & A_6 x + A_9 \\ B_4 x + B_8 & B_6 x + B_9 \end{vmatrix}, \quad \begin{vmatrix} B_4 x + B_8 & B_6 x + B_9 \\ C_4 x + C_8 & C_6 x + C_9 \end{vmatrix},$$

$$\begin{vmatrix} C_4 x + C_8 & C_6 x + C_9 \\ A_4 x + A_8 & A_6 x + A_9 \end{vmatrix}. \quad (27)$$

Equations (26) and (27) are obvious. Equation (25) can be obtained by considering the triangularized matrix (8), which can be of configurations IV, VI, or VII in (9). Since roots of G_x are roots of R_x , it follows that G_x divides R_x . Strictly speaking, rather than R_x , R_x/G_x is the desired intersection polynomial since it has no extraneous roots.

These extraneous x values should not be confused with intersections at infinity, which are solutions of the homogeneous quadric equations (12) with $w = 0$. The following example amply illustrates this fact:

$$\begin{aligned} x^2 + y + 2z &= 0 \\ xy + 2z &= 0 \\ zx + y + z &= 0. \end{aligned}$$

The x, y, z intersection polynomials are, respectively, $x^2(x + 2), 4y^2(3y + 4), z^2(3z + 4)$. An extraneous root $x = 1$ must be removed from the resultant of the x -homogeneous equations. This extraneous root is not a solution at infinity because such solutions necessarily have $x = 0$ (this is obvious by examining the first equation). The extraneous factors in the resultant of the x -homogeneous equations always indicate that $(w : x : y : z) = (0 : 0 : y : z)$, where y, z are not both zero, is an intersection of the given quadrics. But intersections at infinity do not imply that there will be extraneous factors; for example, (20), (21) always have intersections at infinity.

5. AN EMPIRICAL EVALUATION OF THE METHOD

To find out whether this use of multivariate resultants leads to a practical algorithm, we incorporated the method into a solid modeling system running on a Silicon Graphics IRIS 4D/60 workstation and performed an empirical

evaluation.⁴ The performance of our implementation is reasonable but not perfect: for 150 3-quadric configurations involving 650 intersections, 663 intersections are found; among them, 13 intersections are bogus and 29 intersections are not accurate enough (one or more coordinates of an intersection has less than 6 correct digits). Thus over 90 percent of the values computed by the algorithm were correct to better than 6 digits. This outcome is rather satisfactory given that determinant expansions and GCD computations have to be performed using floating-point arithmetic with limited precision. A detailed description of the experiment and an analysis of our results follow.

5.1 Empirical Results

The typical situations under which purely algebraic methods for curve and surface intersections have been known to fail in practice involve various types of degeneracies such as tangencies or reducible intersections (i.e., intersection curves which are a pair of conic sections or a line plus a space cubic). Such conditions are difficult to detect and handle reliably when armed only with algebraic coefficients for the curves and surfaces.

To understand how our method works in practice, we examined a number of test cases, representative examples of which are shown in Figures 1–4. In Figure 1, the cone and sphere intersect in a degenerate nodal space quartic. The cylinder intersects this quartic in four points. There are two degeneracies illustrated in Figure 2(a). The two cylinders intersect in a pair of ellipses, and the horizontal cylinder intersects the cone in a line and a space cubic. In Figure 2(b) we remove the first of these degeneracies by shrinking the size of the vertical cylinder. Then in Figure 2(c) we remove the second degeneracy by translating the horizontal cylinder down by a small amount. Figure 3 also has a degeneracy in that the two cylinders intersect in a pair of straight lines. Finally, the cone and cylinder of Figure 4 intersect in a nondegenerate space quartic with two branches, and the sphere intersects both the cone and the cylinder in nondegenerate one-branch space quartics.

Correct results were generated for these and all other cases we tried. The points of intersection computed by our algorithm are illustrated in the figures by white \times 's. The intersection curves from two of the three pairs of quadrics are also shown in yellow and cyan, respectively.

We wanted to understand how sensitive the calculations were to small perturbations in the coefficients. Furthermore, this method introduces a certain coordinate axis bias since it tries to find roots along one axis at a time. Our approach to understanding the sensitivity while minimizing the effects of coordinate axis bias is given below. In the following we speak of geometric versus algebraic descriptions. The former representation is based on storing, for example, a center point and radius for a sphere. The latter is

⁴In our evaluation only natural quadrics are used, since another intersection algorithm is needed to verify the results. This should not be construed to mean that the algorithm works only for natural quadrics.

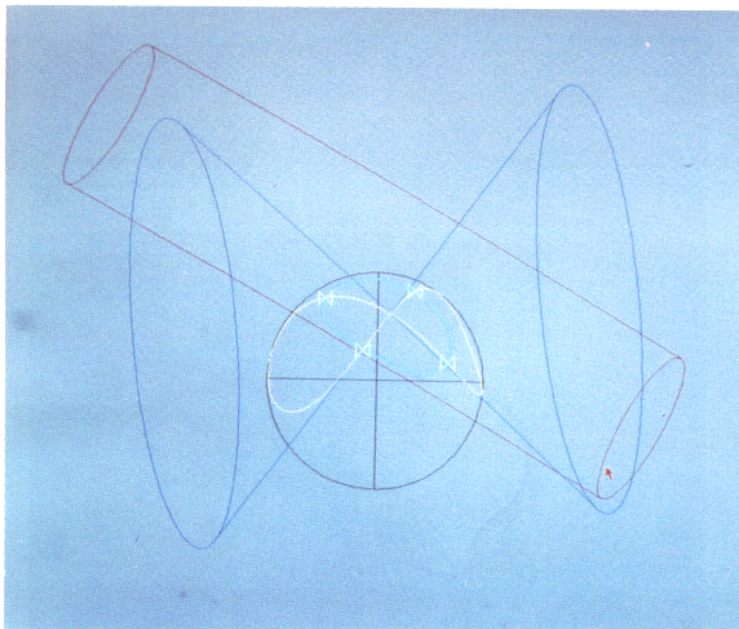


Figure 1

based on storing the 10 coefficients of the general second degree equation in x , y , and z . See Miller [12] for details.

For each of the cases illustrated in Figures 1-4, a series of tests were performed. The i th test did the following:

1. Copy the original geometric data describing the quadrics to Q_1 , Q_2 , and Q_3 .
2. Generate the algebraic representations from the geometric ones, and intersect the 3 quadrics using the algorithm described in this paper. Save the results.
3. Generate i random rotation axes and i random angles in the range $-\pi$ to $+\pi$. Save the axes and angles.
4. for $j = 1$ to i do

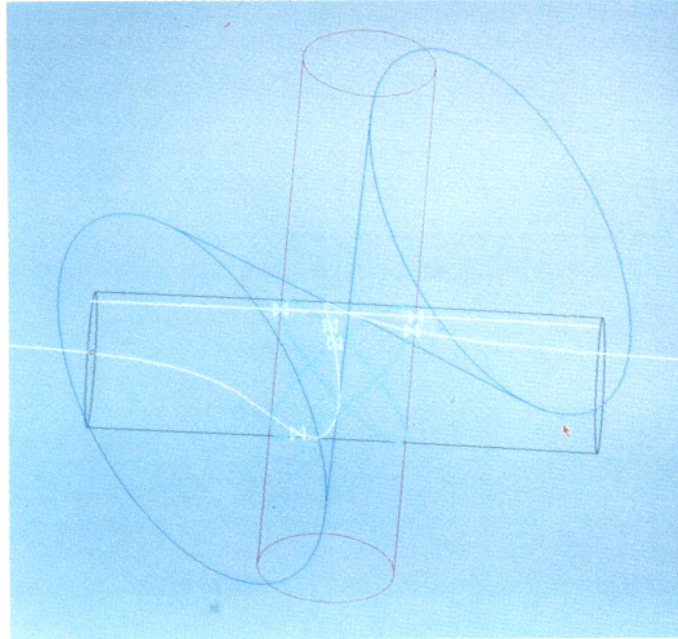
$$Q_k = \text{rotation } j \text{ applied to } Q_k, k = 1 \text{ to } 3.$$

5. Generate the algebraic representations from the (transformed) geometric ones and intersect the 3 quadrics using the algorithm described in this paper. Save the results.
6. for $j = i$ downto 1 do

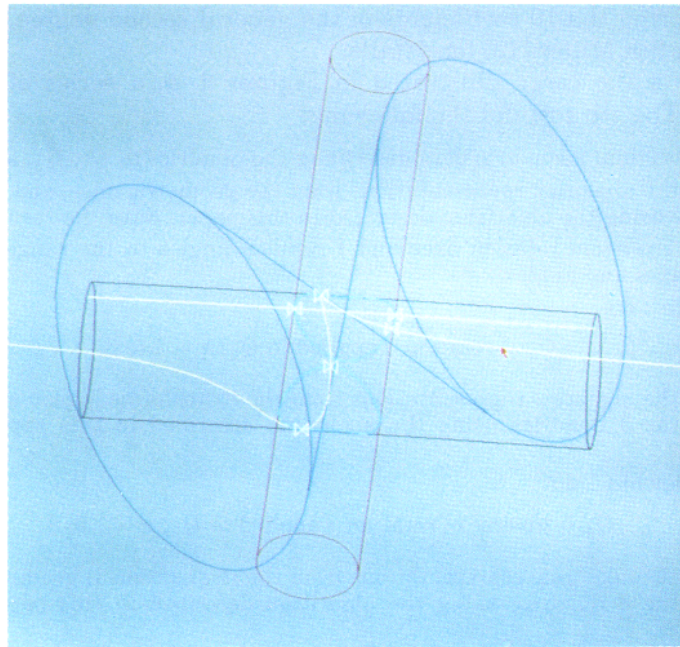
$$Q_k = \text{inverse of rotation } j \text{ applied to } Q_k, k = 1 \text{ to } 3$$

7. Generate the algebraic representations from the (transformed) geometric ones and intersect the 3 quadrics using the algorithm described in this paper. Save the results.

Note that we perform all basic modeling and editing on the geometric representation, and we only convert to the algebraic form when we are ready to apply the algorithm of this paper. That is why the test procedure applied

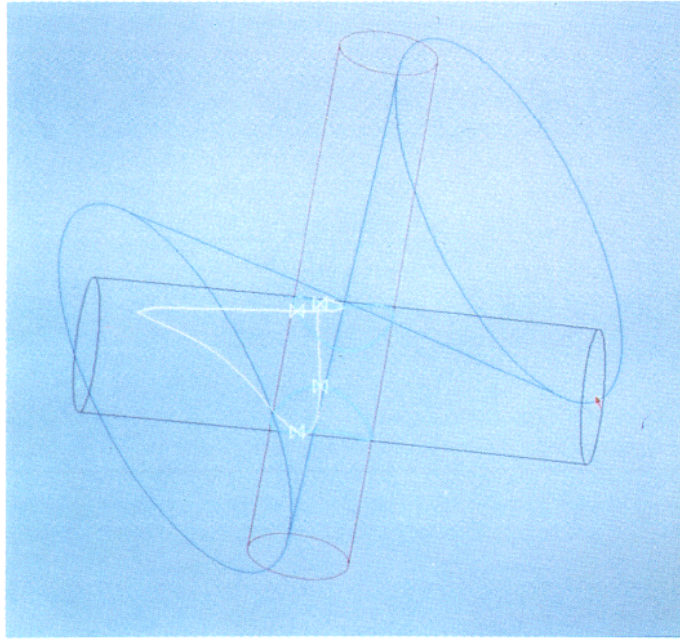


(a)



(b)

Figure 2



(c)

Fig. 2. (Continued)

the rotations to the geometric representations instead of to the algebraic ones. We believe this approach is reasonable since it exactly mirrors the overall modeling environment in which such an algorithm would exist. Experience has shown that database representations and modeling operations based on geometric descriptions are far more robust than ones built on algebraic representations [7].

For the i th test at steps 2, 5, and 7, there were never any errors at steps 2 or 7; the results at step 5, however, were occasionally incorrect. There were no missing intersections. Nevertheless two types of errors did occur when potential intersections were computed but deemed not to lie on the quadric surfaces when substituted back into the implicit equations: these intersections were either bogus or insufficiently accurate. The number of times these errors occurred at step 5 for the given test cases is summarized in Table I. As expected, the situations involving degeneracies generally fared the worst.

5.2 Analysis of the Empirical Results

A major concern in our implementation is the use of floating-point GCD computations, which are known to be highly unstable. In our implementation, there are two occasions where GCD computations are needed: removing extraneous factors from the multivariate resultant R_x (or R_y , R_z) and solving three equations in two variables; that is, after one of the variables x , y , z is calculated, we need to solve for the other two variables using the original three quadric surface equations. The GCD computations for

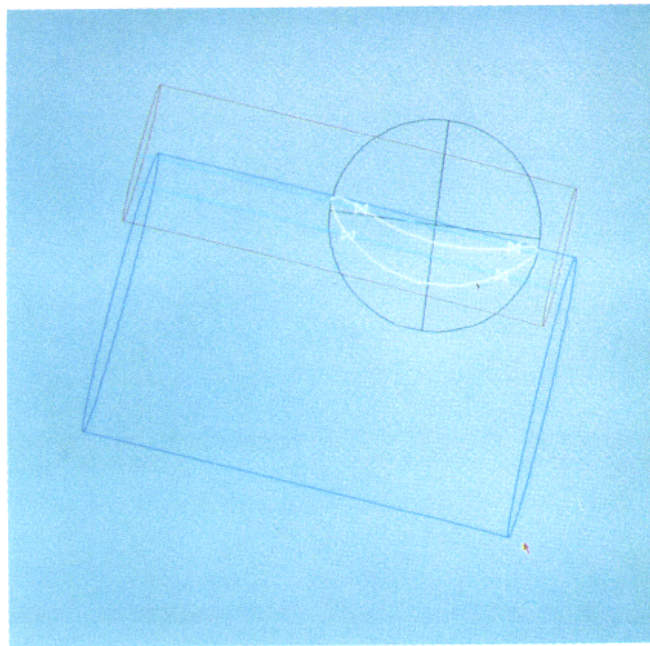


Figure 3

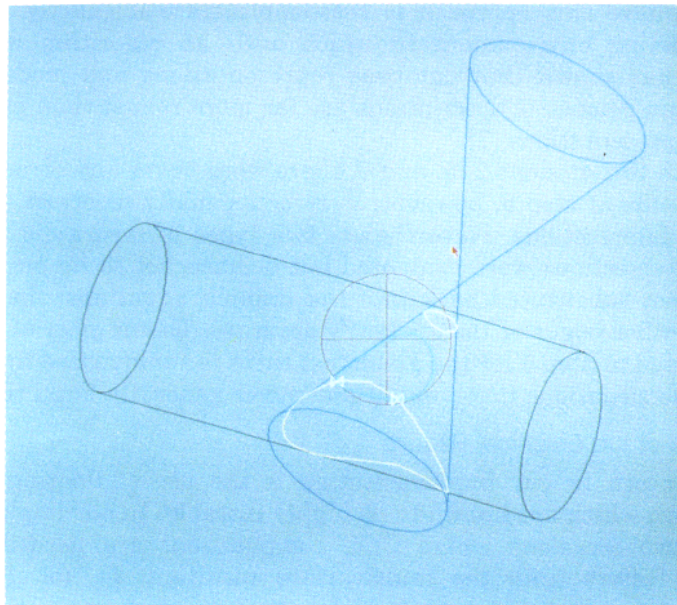


Figure 4

Table I. Summary of Errors at Step 5

Figure number	Degeneracies	Number of intersections			Total errors
		Total	Inaccurate	Bogus	
Fig. 1	Nodal space quartic	100	0	0	0
Fig. 2(a)	2 ellipses and line + space cubic	150	13	3	16
Fig. 2(b)	Line + space cubic	150	11	5	16
Fig. 2(c)	None	100	4	4	8
Fig. 3	2 lines	100	0	0	0
Fig. 4	None	50	1	1	2

Note. There are no errors at Steps 2 and 7.

extraneous factors can be done with a high degree of accuracy, since the two polynomials involved are arrived at with very little computation (see (25), (26), and (27)). Furthermore, in practice, there may not be any extraneous roots—for example, when the rank of matrix (8) is 3. The same cannot be said for the GCD computations involved in solving three equations of two variables. The roots obtained from the intersection polynomial certainly do not have the same degree of accuracy as the coefficients of the quadric equations since the intersection polynomial is obtained after some determinant expansion; consequently the coefficients of the three quadric equations in the two remaining variables are not as accurate as the given quadric surface coefficients. But GCD computations in this context do not require as much accuracy, for we know that the GCD has to be a nonconstant polynomial; that is, we only need to identify nearby roots of two polynomials.

Thus our algorithm finds all the intersection points, at the price of admitting some bogus intersections of coordinates that are not sufficiently accurate (in our experiments this means less than 6-digit accuracy). Bogus intersections can easily be detected; they are simply too far from one or more of the quadrics. Intersections with inaccurate coordinates can easily be polished with Newton-Raphson iteration. In our implementation, we simply applied Newton-Raphson to any point in error (that is, bogus or inaccurate). The bogus roots converged to actual intersection points, and the inaccurate points were refined. This refining procedure is qualitatively different from using Newton-Raphson for equation solving. An average of two or three iterations were sufficient to refine an inaccurate root to an acceptable value. This treatment of bogus roots required a final filtering operation to make sure that roots were not recorded multiple times, but allowed us to avoid having to distinguish between inaccurate and bogus roots. Supplemented by this Newton-Raphson refinement, then, the method has yet to fail in any of our experiments.

6. CONCLUSION

To best address the practicality of the techniques described in this paper, we must again consider the larger context in which they are to be applied. In particular, recall step 1 of the Boundary Evaluation algorithm described in Section 1. We first compute the unbounded intersection curves between the

unbounded quadric surfaces on the model. These curves are then partitioned into pieces such that the interior points on each piece will have an identical classification (INSIDE, OUTSIDE, ON) with respect to all the other quadric surfaces on the model.

A robust implementation which works at least in the context of the natural quadrics is as follows. Compute the underlying quadric surface intersection curves using the geometric methods of Miller [13], O'Connor [17], or Piegl [18], detecting conic sections, isolated intersection points, and other reducible cases in the process. When partitioning these curves, use robust geometric schemes whenever possible. Certainly straight lines and conics can be partitioned in this fashion Miller [12]. The straight line plus space cubic case can be addressed by substituting the parametric representation of the cubic into the implicit representation of the quadric and solving the resulting degree 6 polynomial in the curve parameter. Only when all of the pairwise intersection curves between the 3 quadrics are nondegenerate quartic curves (a condition which will have been detected geometrically "by default" since none of the previous cases will have been discovered) are the algebraic schemes of this paper invoked. Our experience indicates that the application of these methods in that framework is highly reliable.

The techniques described in this paper (as well as the other conic curve and quadric surface intersection schemes surveyed in the introduction) are applicable in any modeling system which has conics and quadrics as one of its geometric forms. As with any nontrivial software application, the architecture of modeling systems ought to be layered so that high-level modules are unaware of the actual curve and surface representations. When such a high-level module requires the intersection of two geometric entities, it ought simply to call a low-level intersection utility that examines the types of the two objects and invokes an appropriate intersection algorithm. If, for example, this low-level routine receives two natural quadrics, then the relevant quadric-quadric intersection algorithm can be invoked. If it receives a quadric and a rational B-spline, then it might need to convert the quadric to a rational B-spline and invoke the routine that intersects two rational B-splines.

This architectural philosophy is especially important for the robust practical implementation of the technique described in this paper. Suppose (e.g., during the course of the Boundary Evaluation algorithm) that a nondegenerate 4th-degree quadric surface intersection curve (c) is generated from the intersection of quadrics Q_1 and Q_2 . Suppose further that it must now be partitioned at its points of intersection with another quadric surface (Q_3). The algorithm of choice depends upon the types and relative geometric orientations of the three quadrics. For example, it may be the case that Q_1 intersects Q_3 in a degenerate curve of some sort. If so, it will be more reliable numerically to compute these degenerate intersection curves and then intersect them with Q_2 than it would be to do a 3-quadric intersection. On the other hand, if there are no degeneracies between any of the 3 quadrics, then the methods of this paper are numerically reliable and ought to be used.

There is another aspect to this claim of "broader than quadrics" applicability. The method of multivariate resultants can, in theory, be applied to

higher order surfaces. We chose to study quadric intersections for several reasons:

- (1) to test the method on the simplest nonplanar case;
- (2) the resultant matrices get intractably large (given today's technology) as the surface degree rises;
- (3) there are some nice quadric-specific tricks that allow us to eliminate the denominators (cf. Section 3).

The only real obstruction with respect to using these techniques with higher order surfaces is the size of the matrices involved. As computers get larger and faster, this will become less of a problem. Now that the theory has been developed and tested in the realm of quadrics, we should be able to use it for higher order surfaces when the requisite computer hardware technology reaches a suitable level.

The practical applicability of this approach has been established. It provides a solution when nonnatural quadrics are involved; for natural quadrics, it provides a plausible alternative to existing algorithms.

ACKNOWLEDGMENTS

We would like to thank John Canny of the University of California, Berkeley, for informing us of Macaulay [9].

REFERENCES

1. BAJAJ, C., GARRITY, T., AND WARREN, J. On the applications of multi-equational resultants. Tech. Rep. CSD-TR-826, CAPO Rep. CER-88-39, Computer Science Department, Purdue University, West Lafayette, Ind., Nov. 1988.
2. CANNY, J. *The Complexity of Robot Motion Planning*. M.I.T. Press, Cambridge, Mass., 1988.
3. CHAR, B. W., GEDDES, K. O., GONNET, G. H., AND WATT, S. M. *Maple User's Guide*. WATCOM Publications Ltd., 1985.
4. CHIONH, E. W. Base points, resultants, and the implicit representation of rational surfaces. Ph.D. dissertation, University of Waterloo, Waterloo, Ont., Canada, 1990.
5. COLLINS, G. E. The calculation of multivariate polynomial resultants. *J. ACM* 18, 4 (Oct. 1971), 515-522.
6. FAROUKI, R. T., NEFF, C. A., AND O'CONNOR, M. A. Automatic parsing of degenerate quadric-surface intersections. *ACM Trans. Graph.* 8, 3 (July 1989), 174-203.
7. GOLDMAN, R. N. Two approaches to a computer model for quadric surfaces. *IEEE Comput. Graph. Applications* 3, 6 (Sept. 1983), 21-24.
8. LEVIN, J. Mathematical models for determining the intersections of quadric surfaces. *Comput. Graph. Image Process.* 11, 1 (Sept. 1979), 73-87.
9. MACAULAY, F. S. Note on the resultant of a number of polynomials of the same degree. *Proc. London Math. Soc.* (June 1921), 14-21.
10. MACAULAY, F. S. On some formula in elimination. *Proc. London Math. Soc.* (May 1902), 3-27.
11. MACAULAY, F. S. *The Algebraic Theory of Modular Systems*. Stechert-Hafner Service Agency, New York, 1964.
12. MILLER, J. R. Analysis of quadric surface based solid models. *IEEE Comput. Graph. Appl.* 8, 1 (Jan. 1988), 28-42.
13. MILLER, J. R. Geometric approaches to nonplanar quadric surface intersection curves. *ACM Trans. Graph.* 6, 4 (Oct. 1987), 274-307.

14. MORGAN, A. P., AND SARRAGA, R. F. *A Method For Computing Three Surface Intersection Points in GMSOLID*. Research Publication GMR-3964, General Motors Research Lab. 1982.
15. MORRIS, J. L. *Computational Methods in Elementary Numerical Analysis*. Wiley, New York, 1983.
16. OCKEN, S., SCHWARTZ, J. T., AND SHARIR, M. Precise implementation of cad primitives using rational parameterizations of standard surfaces. In *Solid Modeling by Computers: From Theory to Applications*, M. S. Pickett and J. W. Boyse, Eds., Plenum Press, New York, 1984.
17. O'CONNOR, M. A. Natural quadrics: Projections and intersections. *IBM J. Res. Dev.* 33, 4 (July 1989), 417-446.
18. PIEGL, L. Geometric method of intersecting natural quadrics represented in trimmed surface form. *Comput. Aided Des.* 21, 4 (May 1989).
19. REQUICHA, A. A. G., AND VOELCKER, H. B. Boolean operations in solid modeling: Boundary evaluation and merging algorithms. *Proc. IEEE* 73, 1 (Jan. 1985), 30-44.
20. SALMON, G. *Lessons Introductory to the Modern Higher Algebra*. G. E. Stechert, New York, 1924.
21. SARRAGA, R. F. Algebraic methods for intersections of quadric surfaces in GMSOLID. *Comput. Vision, Graph. Image Process.* 22, 2 (May 1983), 222-238.
22. VAN DER WAERDEN, B. L. *Modern Algebra*, 2nd ed., Frederick Ungar, New York, 1950.

Received July 1988; revised May 1990; accepted January 1991

Editor: Greg Nielson