

Triangulation and Display of Rational Parametric Surfaces*

Chandrajit L. Bajaj Andrew Royappa[†]
Department of Computer Sciences
Purdue University
West Lafayette, IN 47907
bajaj@cs.purdue.edu

Abstract

We present a comprehensive algorithm to construct a topologically correct triangulation of the real affine part of a rational parametric surface with few restrictions on the defining rational functions. The rational functions are allowed to be undefined on domain curves (*pole curves*) and at certain special points (*base points*), and the surface is allowed to have nodal or cuspidal self-intersections. We also recognize that for a complete display some real points on the parametric surface may be generated only by complex parameter values, and that some finite points on the surface may be generated only by infinite parameter values; we show how to compensate for these conditions. Our techniques for handling these problems have applications in scientific visualization, rendering non-standard NURBS, and in finite-element mesh generation.

1 Introduction

Points on a parametric surface patch can be generated by sampling the parametric functions over some region of the parameter domain. Because of this, the display of patches of polynomial parametric surfaces is well-understood [16, 22]. Some methods address in detail the problem of generating a triangulation on a surface that is sensitive to variations in surface curvature: view-dependent methods [24] as well as view-independent [7].

The parametric functions that define a surface can be viewed as a map from \mathcal{R}^2 into \mathcal{R}^3 . “Domain sam-

pling” methods such as the above assume that the parametric functions are defined and continuous in the region of the parameter domain that is being mapped. If the parametric functions are *rational*, however, they could be undefined at some points in \mathcal{R}^2 . Many surfaces (including simple ones such as some quadrics) are given by rational maps which are undefined at domain points and curves.

We investigate how to correctly triangulate a portion of a rational parametric surface, given a rational map that defines the surface. Our techniques are applicable whether this portion of the surface is described by a bounded region of the parameter domain, or by a bounding box in \mathcal{R}^3 . In this formulation the problem is of interest to CAD designers as well as scientists interested in surface visualization. The former usually express the rational functions defining the surface in terms of the rational Bezier or B-spline bases with non-negative weights, restricting the rational functions to a standard part of the domain. However, researchers are considering non-standard generalizations to rational patches in which the rational functions are not defined everywhere [25], making our techniques relevant. In addition we consider the problem of constructing *triangulations* on singular rational parametric surfaces, especially surfaces that self-intersect. Constructing triangulations on such surfaces is useful for mesh generation in finite-element analysis.

A rational parametric surface is defined by the three rational functions:

$$x(s, t) = \frac{X(s, t)}{W(s, t)}, \quad y(s, t) = \frac{Y(s, t)}{W(s, t)}, \quad z(s, t) = \frac{Z(s, t)}{W(s, t)} \quad (1)$$

where X, Y, Z, W are polynomials with real coefficients and no common factor. Our surface triangulation problem is as follows: Given a bounded region of the domain or a bounding box in \mathcal{R}^3 , compute a topologically correct triangulation of the correspond-

*This research was supported in part by AFOSR grants F49620-93-10138, F49620-94-1-0080, NSF grants CCR 92-22467, DMS 91-01424, NASA grant NAG-1-1473 and ONR grant N00014-94-1-0370

[†]Current Address: Department of Computer Science, University of New Hampshire, Durham, NH 03824

ing portion of the surface defined by (1). In this paper we provide all details for the bounding box case of the problem, noting that our techniques apply as well for the bounded region in the parametric domain. We further require the surface *triangulation* to have all the requisite properties of a simplicial mesh [15], i.e. it must be a triangular mesh whose edges meet only at vertices, without dangling edges, etc.

In this paper we describe the various subproblems that arise in trying to solve the above problem when we don't place any restrictions on the rational functions $x(s, t), y(s, t), z(s, t)$. We then give a solution for each subproblem, and combine the solutions into a comprehensive algorithm. The subproblems are explained in detail in the full version of the paper[9]. They are: domain poles, domain base points, surface singularities, complex parameter values, and infinite parameter values. We mention them briefly here.

1. **Domain poles.** The map yields a divide by zero at points satisfying $W(s, t) = 0$, the poles of the rational functions. These domain poles are algebraic curves. The parametric functions cannot be evaluated at such points and a domain triangulation which intersects these domain pole curves will yield invalid surface triangulations.
2. **Domain base points.** The map is undefined at points satisfying $X(s, t) = Y(s, t) = Z(s, t) = W(s, t) = 0$. There are finitely many such points, called domain *base points*. It is known that a rational curve on the rational parametric surface is the image of each base point however the points on this curve cannot be directly computed using the given rational functions. Ignoring domain base points can lead to a topologically incorrect surface triangulation.
3. **Surface singularities.** The given rational surface may be singular. Even if the rational map has no poles or base points, mapping an arbitrary domain triangulation onto the parametric surface may yield intersecting surface triangles.
4. **Complex parameter values.** Some real points of the surface are generated only by complex parameter values.
5. **Infinite parameter values.** Some finite points of the surface are generated only by infinite parameter values.

Note that for modern day graphics display and NURBS rendering, subproblems (4) and (5) need to

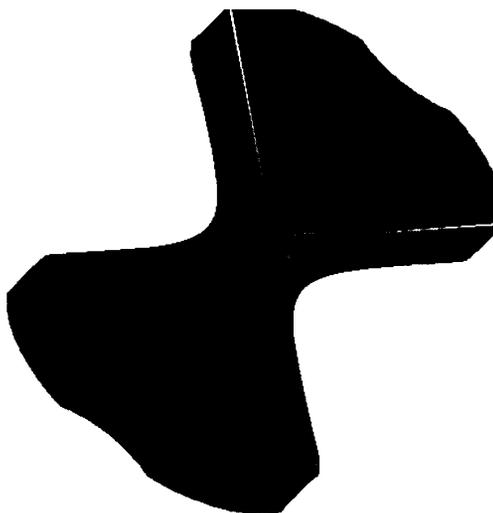


Figure 1: Hyperboloid of 1 sheet with seam curve gaps caused by base points

be addressed while subproblem (3) is not essential (although z-buffering still causes wavy lines along polygon intersections due to aliasing). The above problems can be extended to include rational parametric surfaces and rational varieties in higher dimensions, but we don't discuss this here. The general flavor of the methods discussed will still apply, although implementing higher-dimensional methods would require more computational tools.

The rest of this paper is organized as follows. First in section 2 we discuss two approaches: either directly approximating the surface in the range space of the parametric functions, or computing those portions of the domain that map onto the desired parts of the surface. We argue that the domain-space approach is preferable in this context. Next in section 3 we present techniques for dealing with each subproblem in section 1. In section 4 we then collect these techniques into a comprehensive algorithm for generating topologically accurate surface triangulations and provide practical details and simplifications based on our extensive experimentation.

2 Domain space vs. range space approaches

One way to construct a triangulation of a parametric surface is to evaluate the parametric functions at various points on the parameter domain, and link to-

gether the resulting surface points to form a simplicial mesh. When considering arbitrary rational parametric surfaces, the evaluations of the parametric functions fail at the *poles* and *base* points and techniques of this paper are needed to circumvent them.

Another way to approach the problem is to work directly in the range space of the rational function map. Since we are only interested in portions of a surface inside a bounding box, and poles correspond to surface points at infinity, a range-space method can avoid explicitly evaluating the rational functions at poles. However base points still cause problems. The following system of equations is equivalent to (1): $W(s,t)x - X(s,t) = 0$, $W(s,t)y - Y(s,t) = 0$, $W(s,t)z - Z(s,t) = 0$. One can theoretically implicitize the parametric surface by eliminating s, t from this system [19] using several available methods [21] and then approximate the resulting implicit surface directly. Note that a parametric surface of degree n could have an implicit equation of degree n^2 . However, implicit surface approximation techniques [11, 18] don't handle surface singularities very well, although research is being done to overcome this. Since we would like to display surfaces with complicated singularities and several real sheets, we avoid the range-space approach. We show instead that a careful evaluation of the domain is sufficient to generate an accurate triangulation of the parametric surface.

3 Techniques for overcoming difficulties

In this section we outline the basic idea for solving each of the problems addressed above. The complete algorithm is presented in the next section along with implementation specific details.

3.1 Partition of domain by pole curves

Rational functions are undefined at points in the domain where their denominator vanishes, and continuous everywhere else. Hence, the pole curve partitions the parameter domain into regions, such that inside each (open) region the functions of the parametric map are defined and continuous.

Therefore, our approach to handling pole curves is simple: we partition the domain by the pole curve. In particular, we construct a special triangulation of the domain that respects this partition. In this triangulation, a domain triangle contains pole points only on its boundary and not in its interior. We construct a piecewise-linear topologically correct approximation

of the plane algebraic pole curve [6, 10] and then construct a triangulation which conforms to this linear approximation i.e. the triangles abut the linear curve segments at vertices or edges. The conforming triangulation may require additional points to be inserted in the piecewise linear approximation of the curve. Bounds on the numerical approximation (number of bits of precision) required for a topologically correct linear approximation of an algebraic curve is given in [10] and are based on the gap theorem of [12].

Once such a triangulation is constructed, we know that each domain triangle maps onto a single-sheeted patch, since there are no pole points in the interior (pole points at a vertex correspond to points at infinity, and therefore the patch may be semi-infinite). A conventional domain sampling technique is used in the interior of the triangle to mesh the patch to any desired precision [24]. The patch can then be clipped against a bounding box, if necessary. If base points are not present, domain partitioning combined with the handling of infinite parameter values (discussed below) suffices to generate a topologically correct triangulation of the parametric surface, even if it is multi-sheeted.

3.2 Base points and seam curve parameterizations

When base points are present, it is not sufficient to just handle pole curves as gaps may still be present, as in Figure 1. The surface approximation will then not be topologically correct, since the surface approximation will be "torn" along the seam curves.

To handle base points, we must "stitch" the surface up along seam curves. This can be done in the framework of domain partitioning, as follows. We compute all base points and insert them into the domain triangulation as additional vertices – thus base points will occur explicitly at the vertices of a domain triangle.

In general, approaching a base point along different directions in the domain leads to a different surface point (in the limit). Thus a base point "blows up" onto an entire "seam" curve on the surface [23] – each point of this curve corresponds to a different limit direction at the base point. A consequence of this fact is that a domain triangle with a base point vertex maps onto a four-sided patch on the surface. In general, a triangle with b base point vertices maps onto a $(b + 3)$ -sided patch – a fact exploited in [25] to represent multi-sided patches over triangular domains.

Once we have a *parameterization* of the seam curves, it is easy to generate the patch corresponding

to a domain triangle with base point vertices, however many sides it has. Each of the two edges adjacent to a base point vertex corresponds to a particular direction, and therefore to a particular parameter value. The two parameter values then define a segment of the seam curve. This curve segment is the side on the patch that corresponds to the domain base point.

We now discuss the computation of seam curve parameterizations. Points on a rational parametric surface are given as follows (temporarily using projective coordinates for notational convenience): $\rho X = X(s, t)$, $\rho Y = Y(s, t)$, $\rho Z = Z(s, t)$, $\rho W = W(s, t)$ where ρ is a non-zero constant of proportionality (we still use an affine domain, which is sufficient as we later show).

Then, let O be a common solution of the curves $X = 0, \dots, W = 0$. Furthermore, let us suppose that O is a point of multiplicity q on each of the curves $X = 0, \dots, W = 0$, and that the curves have no common tangent at O . Then the image of the base point O is a rational curve of degree q on the surface [23].

In [13, 14], a method is given to find the parametric equations of this curve. The basic idea is to pass a pencil of lines through the base point and then use the slope of these lines as a parameter, since approaching the base point from each direction leads to a different point on the seam curve. The seam curve equations are not given explicitly, but as quotients of certain polynomials. The algorithm fails when the curves $X = 0, \dots, W = 0$ have common tangents at O ; in this case the parametric equations given by this algorithm generate only a single point of the seam curve. In [20] a method is given for parameterizing seam curves that works for all cases (i.e., even when the tangents are equal). However, it is much more expensive than the previous method and not currently practical: multivariate resultants are used to compute a projection onto a plane of all the seam curves simultaneously, yielding a bivariate equation. Along with the projection, a rational map R is computed between the projection and the curves on the surface. A bivariate factorization algorithm (over the complexes) such as [4] must first be applied to separate out the the projections of the individual curves. Each projected seam curve is then parameterized using a general curve parameterization technique [2], and finally mapped onto the surface using the rational map M .

Our solution to the problem is a simplification of the method of [13] and stated in the Theorem below which allows us to give an explicit formula for the parametric equations of the seam curve and a “blowing up” of the base point based on affine quadratic trans-

formations [1, 6] for the special case when the tangents at the base point are equal. The affine quadratic transformation we use are of the type $T : x = r, y = rs$ with inverse $T^{-1} : r = x, s = y/x$ with the base point translated to the origin. This transformation is applied to the product of the curves $X = 0, \dots, W = 0$ whose tangents at the base point are equal. Its affect is to yield a curve with distinct tangents at the base point. The transformation being birational (i.e. rational and with an inverse which is rational) allows one to map the seam curve parameterization for the transformed domain curves back to the original. Only a finite number of these transformations are required to separate the tangents[1]. Details of its application for tracing through curve singularities is given in [6]. Bounds on the numerical approximation (number of bits of precision) required for the blowing up procedure at the base point are similar to those given for singularity computation in [10] and are based on the gap theorem of [12]. This then together with the explicit formula for the parametric equations of the seam curve given below, solves this base points problem.

THEOREM 1 *Let (a, b) be a base point of multiplicity q . Then for any $m \in \mathcal{R}$, the image of a domain point approaching (a, b) along a line of slope m is given by $(X(m), Y(m), Z(m), W(m)) =$*

$$\left(\sum_{i=0}^q \left(\frac{\partial^q X}{\partial s^{q-i} \partial t^i} (a, b) \right) \binom{q}{i} m^i \right.$$

$$\left. \dots \sum_{i=0}^q \left(\frac{\partial^q W}{\partial s^{q-i} \partial t^i} (a, b) \right) \binom{q}{i} m^i \right)$$

PROOF. See the full version of the paper[9].

COROLLARY 1 *If the curves $X(s, t) = 0, \dots, W(s, t) = 0$ share t tangent lines at (a, b) , then the seam curve $(X(m), Y(m), Z(m), W(m))$ has degree $q-t$. In particular, if $X(s, t) = 0, \dots, W(s, t) = 0$ have identical tangents at (a, b) , then for all $m \in \mathcal{R}$ the coordinates $(X(m), \dots, W(m))$ represent a single point.*

3.3 Partitioning along surface singularities

Earlier, we mentioned two reasons why a domain triangulation might not stay a triangulation when it is mapped onto a parametric surface. The first reason was because the domain sampling density was not high enough, and the second reason was because the surface might self-intersect. The first case can be handled by increasing the domain sampling density (either locally or globally, although local curvature-sensitive

sampling is much preferred since it generates fewer polygons). Several domain sampling techniques already adjust the sampling density due to curvature, so we focus on the second case.

The domain-partitioning technique lends itself to generating triangulations on singular surfaces. The key idea is to compute those points and curves in the parametric domain that map onto surface singularities, and then partition the domain by these points and curves (as well as by the pole curves). If this is done, no domain triangle will contain in its interior a point that map onto a surface singularity. Hence, triangles on the surface will meet only along their edges or at their vertices, even if the surface is singular.

Domain curves (and points) mapping onto surface singularities can be computed by solving systems of polynomial equations [3, 5, 12]. Singularities correspond to domain points where the Jacobian matrix of the rational map does not have full rank. We can compute the symbolic Jacobian matrix and equate its minors to zero, yielding a set of polynomial equations whose common solution are domain points and curves that map onto surface point and curve singularities. Multivariate resultants [19, 5, 21] can be used to project the solutions onto the parameter plane, after which a curve-tracer can be used to compute a linear approximation as for the domain pole curves. For tracing the curve one can use either subdivision methods, e.g. [17], or a marching method such as [6, 10]. Bounds on the numerical approximation (number of bits of precision) required for the singularity tracing computation are given in [10] and are based on the gap theorem of [12].

For example, consider the surface given by the the following equations, taking $x(s, t) = X(s, t)/W(s, t)$, etc.

$$\begin{aligned} X(s, t) &= s^3 + st^2 - 3s \\ Y(s, t) &= (s^2 + t^2)^2 - 3(s^2 + t^2) \\ Z(s, t) &= s^2t + t^3 - 3t \\ W(s, t) &= (s^2 + t^2)^2 + 2(s^2 + t^2) + 1 \end{aligned}$$

This is a surface of revolution (see Figure 3) and has a point singularity at the origin.

It can be shown that the domain points mapping onto the surface singularity satisfy $(t^2 + s^2 - 3)(t^2 + s^2) = 0$. Thus the circle of radius $\sqrt{3}$ centered at the origin, and the origin itself both map onto the surface singularity at $(0, 0, 0)$. This circle and the origin, partition the parameter domain into regions that meet at the surface self-intersection. By partitioning the parameter domain by treating the curve $t^2 + s^2 - 3 = 0$

and the point $(0, 0)$, similar to the domain pole curves, we can construct a triangulation on this surface.

3.4 Computing complex parameter values

We now show one way to compute the complex parameter values that map onto these points. Let the parameters s, t denote complex numbers given as $s = a + bi$, $t = c + di$, where $a, b, c, d \in \mathcal{R}$ and $i = \sqrt{-1}$.

Then the parametric map from $\mathcal{C}^2 \rightarrow \mathcal{R}^3$ can be expressed as

$$\begin{aligned} x(s, t) = x(a + bi, c + di) &= X_R(a, b, c, d) + X_I(a, b, c, d) \cdot i \\ y(s, t) = y(a + bi, c + di) &= Y_R(a, b, c, d) + Y_I(a, b, c, d) \cdot i \\ z(s, t) = z(a + bi, c + di) &= Z_R(a, b, c, d) + Z_I(a, b, c, d) \cdot i \end{aligned}$$

where X_R denotes the real part of $x(a + bi, c + di)$ and X_I denotes its imaginary part, etc.

Then $X_I(a, b, c, d) = 0$, $Y_I(a, b, c, d) = 0$, $Z_I(a, b, c, d) = 0$ form a system of three equations in four unknowns whose solutions give parameter values that map to real surface points. In general, such a system has a one-dimensional solution set.

Note that this particular system has the trivial two-dimensional solution $b = d = 0$ which must be excluded. Thus the marching methods of [6] cannot be used directly; rather, as for surface singularities, we must use resultants to first compute a projection of the space curve. After deleting the extraneous component due to the trivial solution, we can trace the projected plane curve and finally map it onto the space curve using the inverse of the projection.

The points (a, b, c, d) of the space curve give complex parameter values $s = a + bi$ and $t = c + di$ that map onto real points of the surface.

3.5 Mapping infinity using projective reparameterization

To handle infinite parameter values, we use projective reparameterizations [8]. Specializing Theorem 1 of [8], we use four reparameterizations of the original rational map, given by

$$\begin{aligned} s &= \pm \frac{u}{1 - u - v} \\ t &= \pm \frac{v}{1 - u - v} \end{aligned}$$

Each reparameterized map needs to be sampled only over the unit triangle of its domain ($u \geq 0$, $v \geq 0$, $u + v \leq 1$), yielding a triangular patch. The patches meet along their boundaries and together cover the

entire surface (including finite points that were generated by infinite parameter values in the original surface). Figure 6 shows a member of the Steiner surface family mapped using four reparameterizations. Each piece is the image of a different domain quadrant under the original rational map. Each piece is given a different color.

4 The Triangulation Algorithm

As is common in domain sampling techniques, a triangulation of the parametric domain is mapped onto the surface, yielding a piecewise-linear approximation to it. Triangular surface elements have several advantages, described in detail in [24].

The pole curve partitions the parameter domain into several regions. The rational functions of the map are continuous inside these open regions, and therefore each region maps to a possibly infinite but *single sheeted* surface patch. The algorithm uses the pole curve and any self-intersection image curves to partition the domain into regions. This is done by generating a piecewise-linear approximation to the pole curve and the surface singularity domain points and curves. A conforming triangulation is then constructed of the curve points, base points and sufficiently many other ordinary domain points. The conforming domain triangulation “respects” the pole curves and self-intersection image curve. In other words, the interior of a domain triangle in this triangulation maps onto a patch of the surface that is single-sheeted and does not intersect itself. Figure 2 shows a domain triangulation respecting a pole curve. Each domain triangle is allowed to have up to two vertices that are on the pole curve (note that base points are also on the pole curve). Each domain triangle then corresponds to a single-sheeted surface patch.

We first present the algorithm, and then provide implementation details and examples.

1. (RESTRICT TO FINITE DOMAIN) Perform a projective reparameterization so that the entire surface is mapped in four pieces, each over the “unit” triangle spanned by $(0, 0)$, $(0, 1)$, $(1, 0)$. Treat the four new mappings independently, and for each mapping perform the following steps.
2. (GENERATE POLE POINTS) Compute a piecewise-linear approximation to the pole curve of the current mapping inside the unit triangle.
3. (GENERATE SURFACE SINGULARITY IMAGE POINTS) Compute a piecewise-linear ap-

proximation to the image curves or points of any surface singularity

4. (GENERATE BASE POINTS) Compute all the base points of the current mapping that lie inside the unit triangle.
5. (GENERATE DOMAIN POINTS) Generate points in the rest of the unit triangle according to some fixed or adaptive sampling scheme.
We label each kind of point accordingly as pole, singularity, base or ordinary domain point.
6. (TRIANGULATE) Compute a triangulation of the points thus generated. If the edge of any triangle crosses the pole curve or the singularity image curves, insert the intersection points; if any triangle has three pole vertices, insert its centroid.
7. (MAP TRIANGLES) Every triangle can now have up to 2 pole vertices or base point vertices. Map each triangle onto a surface patch and clip it against the bounding box. Various types of patches result depending on the labels of a domain triangle. They are as follows:

- All vertices are ordinary. The image of the triangle is a finite triangular patch.
- One vertex is a pole. The image is an infinite triangular patch with one corner at infinity.
- Two vertices are poles. The image is an infinite triangular patch with two corners at infinity.
- One vertex is a base point. The base point blows up to a curve on the surface. Approaching the base point vertex along each of its incident edges leads to a different surface point on this curve. Thus, the image is a finite *rectangular* patch.
- Combinations of ordinary, base points, and pole points. The resulting patch can be finite or infinite, with up to six sides.

Mapping each domain triangle is accomplished by walking along its boundary and checking the vertex labels. For clipping, an iteration must be used to locate the intersection(s) of each edge of the surface patch with the bounding box. To map domain triangles with base point vertices, a parameterization of the corresponding seam curve is necessary, as explained previously and elaborated below.

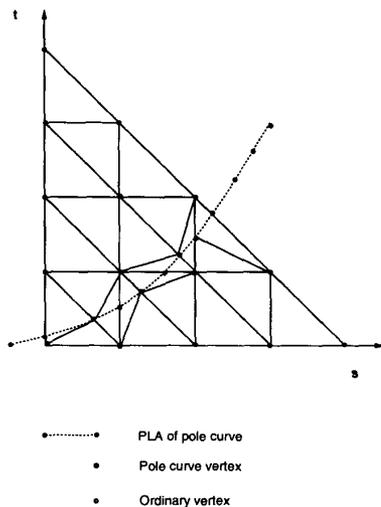


Figure 2: Partition of the domain triangulation by a pole curve

Finally, we could compute surface points corresponding to complex parameter values as explained earlier. Since these points form a one-dimensional family in general (i.e. they are curves), we don't include them in the triangular mesh; they should however be used to augment the surface display in scientific visualizations.

4.1 Details and Implementation Issues

We now discuss the steps in more detail.

In step 2, we use an adaptive tracing scheme based on [6] and suign estimates of *Lipschitz constants* as in [24].

In step 4 we find the base points by solving the equations $F_i = 0, i = 1, \dots, 4$. This is done by picking two of the equations, finding their common solutions, and then checking whether these are solutions of the other two equations. In [3] and [13] a method based on resultants and birational maps is given for finding all base points and their multiplicities directly.

In step 5, points on the unit triangle can be generated either uniformly or adaptively spaced. Points that are uniformly spaced in the s and t directions are easily generated. For instance we can generate $n(n+1)/2$ points by taking $i+1$ equally spaced points on each line $s+t = i/(n-1), i = 0, \dots, n-1$. The points can also be selected based on local surface curvature, for instance using heuristics that test the local "flatness" of the surface using the tangents at triangle

endpoints.

For step 6, any triangulation method can be used, although some triangulations such as the Delaunay have convenient properties [15]. Furthermore for Delaunay triangulations upper bounds are known for the number of additional points required to make the triangulation conform to the linear approximation of the domain pole and singularity curves.

Step 7 is complicated only because of the many cases involved. We know that each of the current domain triangles maps onto a single-sheeted patch with up to 5 sides. The domain triangle is subdivided if necessary, using an adaptive domain sampling technique [24] where again estimates of *Lipschitz constants* are used to decide when a portion of a surface is sufficiently linear to be approximated by a triangular facet.

Finally, base point vertices need special treatment. A domain triangle with b base point vertices maps onto a patch with $b+3$ sides. Three sides of the patch are the images of the domain triangle's three edges, and therefore tracing these sides (for clipping) is not a problem. How to trace the other b sides of the patch is not obvious, and we describe that here. Consider a triangle with a base point vertex \mathbf{p} . Suppose \mathbf{p} is incident to the edges e_1 and e_2 . Let the slope of the edges e_1 and e_2 be m_1 and m_2 respectively. Approaching \mathbf{p} along the line of slope m_1 leads to one point on the surface, and approaching it along the line of slope m_2 leads to another point on the surface. Both these points lie on the seam curve corresponding to the base point. Parameterize the seam curve in terms of m , the slope of lines through the base point. Then the side of the patch that corresponds to the base point vertex can be traced by evaluating the seam curve parameterization at values between m_1 and m_2 .

We have implemented the entire algorithm and tried various techniques for solving the above problems. Based on our experiments we feel that the domain partitioning approach is most suitable. Our implementation is in the Ganith system [7] and was used to generate Figures 1, 3, 6, 4, 5, 7, and 8. The last four show various self-intersecting and multi-sheeted rational parametric surfaces. More information on the software is available via anonymous ftp from ftp.cs.purdue.edu and via World Wide Web access using Mosaic from <http://www.cs.purdue.edu/research/shastra/shastra.html>.

References

- [1] Abhyankar, S. S., (1983) Desingularization of Plane Curves, *Proc. of the Symp. in Pure Math-*

(See color plates, page CP-6.)

- ematics*, 40, 1, 1-45.
- [2] Abhyankar, S. S., and Bajaj, C., (1988), Automatic Parameterization of Rational Curves and Surfaces III: Algebraic Plane Curves, *Computer Aided Geometric Design*, 5, 309 - 321.
 - [3] Bajaj, C., (1990) "Geometric Computations with Algebraic Varieties of Bounded Degree", Proc. of the Sixth ACM Symposium on Computational Geometry, Berkeley, California, 148-156.
 - [4] Bajaj, C., Canny, J., Garrity, T. and Warren, J., (1989), "Factoring Rational Polynomials over the Complexes," *Siam J. on Computing*, 22, 2, 318 - 331.
 - [5] Bajaj, C., Garrity, T. and Warren, J., (1988), "On the Applications of Multi-Equational Resultants," Technical Report CSD-TR-826, Department of Computer Sciences, Purdue University.
 - [6] Bajaj, C., Hoffmann, C., Hopcroft, J., and Lynch, R. (1988), "Tracing Surface Intersections," *Computer Aided Geometric Design*, 5, 285-307.
 - [7] Bajaj, C., and Royappa, A., (1990) "The Ganith Algebraic Geometry Toolkit", in *Proc. of the 1st Annual Conference on the Design and Implementation of Symbolic Computation Systems*, LNCS 429, Springer-Verlag, Capri, Italy, 268-269.
 - [8] Bajaj, C., and Royappa, A., (1993), "Finite Representations of Real Parametric Curves and Surfaces," *Modeling in Computer Graphics*, ed. by B. Flacidierno, T. Kunii, Springer Verlag, Genova, Italy, 347 - 358.
 - [9] Bajaj, C., and Royappa, A., (1993), "Topologically Correct Approximations of Arbitrary Rational Parametric Surfaces", Technical Report CSD-TR-93-009, Department of Computer Science, Purdue University.
 - [10] Bajaj, C. and G. Xu, (1994), NURBS Approximation of Surface-Surface Intersection Curves, *Advances in Computational Mathematics*, 2, 1, 1-21.
 - [11] Bloomenthal, J., (1988) "Polygonization of Implicit Surfaces.", *Computer Aided Geometric Design*, 5, 341 - 355.
 - [12] Canny, J., (1987) "The Complexity of Robot Motion Planning", ACM Doctoral Dissertation Award, The MIT Press, Cambridge, MA.
 - [13] Chionh, E., (1990), "Base Points, Resultants, and the Implicit Representation of Parametric Surfaces." Ph.D. Thesis, University of Waterloo.
 - [14] Chionh, E., and Goldman, R., (1992), "Degree, Multiplicities and Inversion Formulas for Rational Surfaces using U-Resultants", *Computer Aided Geometric Design*, June 1992.
 - [15] Edelsbrunner, H., (1987), *Algorithms in Combinatorial Geometry*, Springer-Verlag.
 - [16] Foley, J., Van Dam, A., Feiner, S.K., and Hughes, J.F., (1990), *Fundamentals of Interactive Computer Graphics*, 523-529, Addison Wesley Publishing Co.
 - [17] Geisow, A., (1983), "Surface Interrogations", Ph.D. Dissertation, School of Computing and Accountancy, University of East Anglia.
 - [18] Hall, M., and Warren, J., (1988), "Adaptive Polygonization of Implicitly Defined Surfaces," in *IEEE Computer Graphics & Applications*, 10, 6, 33-42, November.
 - [19] Macaulay, F.S., (1902), "Some Formulae in Elimination," in *Proc. London Math. Soc.*, 35, 3-27, 1902.
 - [20] Manocha, D., (1992), *Algebraic and Numeric Techniques in Modeling and Robotics*, Ph.D. Thesis, University of California at Berkeley, 1992.
 - [21] Manocha, D., and Canny, J. F., (1991), "Efficient Techniques for Multipolynomial Resultant Algorithms," in *Proceedings of International Symposium on Symbolic and Algebraic Computation*, 271-276, 1991.
 - [22] Rockwood, A., Heaton, K., and Davis, T., (1989), "Real-Time Rendering of Trimmed Surfaces", in *Proceedings of ACM SIGGRAPH '89*, Boston, 107-116.
 - [23] Semple, J.G., and Roth, L. (1985), *Introduction to Algebraic Geometry*, Oxford University Press, 1985.
 - [24] Von Herzen, B., (1989), *Applications of Surface Networks to Sampling Problems in Computer Graphics*, Ph.D. Thesis, California Institute of Technology.
 - [25] Warren, J., (1992), "Creating Multisided Rational Bezier Surfaces Using Base Points", in *ACM Transactions on Graphics*, 11, 2, 127-139.