Hidden Curve Elimination of Trimmed Surfaces Using Bézier Clipping

Tomoyuki Nishita, Shinichi Takita, and Eihachiro Nakamae

ABSTRACT

Shaded display is popular for displaying curved surfaces because of the realistic appearance of the images produced. In the field of computer aided geometric design, however, line drawings are more commonly used. For accurate evaluation of the curved surfaces in these designs, the conventional method of approximating curved surfaces with polygons is insufficient. In this paper we will present a method of hidden line elimination without resorting to polygonal approximation. The method provides the designer smoother, more precise iso-parametric curves than traditional polygonal approximation methods. In addition, the method can draw precise silhouette curves, trimming curves, drawings of patterns represented by curves on the surface, and contour lines. Furthermore, the method is valid even for surfaces which penetrate each other.

Keywords: Hidden curve elimination, Bézier clipping, extraction of silhouette, penetrated surfaces, contour lines, trimmed patch, curve-curve intersection.

1 INTRODUCTION

Line drawings of curved surfaces have traditionally been generated using polygonal approximations and removing hidden lines from the polygons. However, because the displayed edges of the polygons are straight lines, the displayed curves are not smooth, and the shape of the silhouette of an object viewed from a viewpoint is inaccurate. Curves can be displayed more smoothly by increasing the number of polygons used for the approximation, but this results in enormous memory and computational expense.

Surfaces expressed as NURBS or other representations can be converted into rational Bézier surfaces. Therefore in this paper, without loss of generality, we treat surfaces expressed in the rational Bézier form. Curved surfaces are commonly inter-penetrating (to express it from another point of view, it is very difficult to define surfaces which do not intersect). Thus, the surfaces addressed in this paper are permitted to intersect. Furthermore, trimming curves representing holes, geometric patterns, and contour lines on the curved surfaces are also addressed.

Hidden line removal algorithms for curved surfaces present many difficulties which are not present when dealing only with polygonal data. These problems include, for example, testing for intersection of curves on the projection plane (2D curve/curve intersection), extraction of silhouettes, ray/surface intersection to test for visibility of a point on a curve, and the penetration test. The authors have developed a ray tracing [Nishita 1990] and a scan line [Nishita 1991] algorithm for trimmed rational Bézier patches. In this paper we demonstrate a hidden line removal algorithm which uses *Bézier Clipping*, a method developed in conjunction with that work. Bézier Clipping is iterative, and based upon the Bézier curve's convex hull. The intervals of solutions of a polynomial function expressed by the Bézier curve are reduced by clipping the curve, and the process iterates until it converges upon the solutions. This method is more robust than Newton's method.

Newton's method requires determination of a suitable initial value, and it is difficult to guarantee finding all solutions. Bézier Clipping can be used to solve for intersections of pairs of curves or surfaces, of curves or lines with surfaces, and to extract silhouettes. Curves (such as iso-parametric curves, silhouettes, or trimming curves) on the projection plane are expressed as higher order rational Bézier curves (e.g., 3 to 18 degree curves). The Fat Line Method [Sederberg 1990a], based on Bézier Clipping, obtains the intersection points robustly and efficiently (only 2 or 3 iterations are needed).

The rest of the paper is organized in the following manner: In section 2, a brief overview of the previous work is given. The basic concept of our method is explained in section 3. Our intersection test on projection plane is mentioned in section 4. The procedure of extracting curved surfaces overlapping with a curve is described in section 5. How to calculate the intersection curve between two curved surfaces is explained in section 6. The procedure of extracting silhouette curves using Bézier Clipping is discussed in section 7. The processing of trimming curves and pattern curves on curved surfaces is mentioned in section 8. The method displaying contour lines is described in section 9. To show the usefulness of the proposed method, some examples are illustrated in section 10.

2 PREVIOUS WORK

Many hidden line removal methods for polyhedra have been proposed [Appel 1967] [Loutrel 1970] [Sequin 1985] [Kripac 1985]. As these algorithms apply only to polyhedra, they can only be used for curved surfaces if the surfaces are first subdivided into a large number of polygons.

Polygonal approximation methods for display of curved surfaces also been developed. In these methods a grid of points is established on the curved surface to divide the surface into quadrilaterals which can be treated as planar. The straight edges of the quadrilaterals are displayed as an approximation of the iso-parametric curves of the surface. Ohno[Ohno 1983] proposed a method which subdivides the 3D space into 3D "screen boxes" to discriminate efficiently approximated polygons which overlap on the screen. In addition to memory proportional to the number of surface subdivisions, this method also requires memory proportional to the number of space subdivisions. Li[Li 1988] used 2D subdivisions to determine on-screen overlap of approximated polygons, thus reducing the memory requirement of the algorithm. In both methods visibility is determined for the endpoints of straight lines. Therefore, these methods fail to detect invisibility occurring in the interior region of a line when both endpoints are visible. Both of these methods suffer from iso-parametric curves which are not smooth and silhouettes which are inaccurate when the scene is projected for viewing.

To remove hidden lines from curved surfaces without polygonal approximation, extraction of visible segments, detection of intersections of pairs of curves and of rays and surfaces, and extraction of silhouettes, are important problems which must be solved. Various solutions have been developed. Hornung et al.[9] proposed a hidden line removal method based on the idea of the quantitative invisibility, which determines visible and invisible regions in advance [Hornung 1984]. This method was applied to bi-quadratic surfaces [Hornung 1985], and Newton's method was employed to solve for intersections between curves.

Elber and Cohen [Elber 1990] applied Hornung's technique to nonuniform rational B-splines and extended it to treat trimming curves. Visible regions are pre-calculated in order to reduce the number of intersections which must be solved. This method is not valid for inter-penetrating curved surfaces, an important restriction on its general applicability. Curve intersections are detected by combining Newton's method with the cone test [Sederberg 1990b].

The method which we demonstrate in this paper utilizes Bézier Clipping to robustly obtain all solutions.

To display curved surfaces accurately, silhouettes have to be calculated precisely. Silhouette extraction methods have been developed not only for the hidden line removal method, but also for scan line based hidden surface removal. Whitted[Lane 1980] demonstrated a method in which a silhouette curve is obtained by Hermite interpolation after detecting silhouette points by Newton iteration. However, Newton's method requires initial values, and when a silhouette curve doesn't intersect with the boundary of a curved surface, it is impossible to obtain any silhouette curve. Schweitzer and Cobb's [Schweitzer 1982] method extracts silhouette edges robustly by subdividing a curved surface into subpatches. In this method several silhouette points are obtained by approximating the normal of a surface by a cubic surface, and a silhouette curve is expressed by a piecewise linear curve joining them.

Griffiths[Griffths 1984] and Pueyo and Brunet[Pueyo 1987] divided a curved surface into a grid of points in advance, and extracted silhouette points using the z components of normals at grid points. As linear interpolation is applied in extracting the silhouette points, precision is not sufficient. Furthermore, when more than one silhouette curve lies between grid points, it is impossible to extract silhouette points.

Hornung et al. [Hornung 1985],obtained silhouettes by the marching method, starting from a silhouette point. Unfortunately, when using this method it is difficult to guarantee that all silhouettes have been obtained.

Elber and Cohen[Elber 1990] developed a method in which the curved surfaces are recursively subdivided and the subpatches intersecting silhouettes are detected based on the sign of the z component of the surface normal. In this method, each curved surface is first divided into four subpatches, then the presence or absence of a silhouette in each subpatch can be determined.

In contrast, our method can detect parametric subregions containing silhouettes before any subdivision is performed, and is therefore more efficient. Regions containing silhouettes are extracted geometrically using a hodograph (the first derivative of the surface).

References [Hornung 1985] [Elber 1990], state that calculation of ray/surface intersections is computationally expensive, and to avoid unnecessary intersection tests the visible regions have to be determined in advance. Our method obtains ray/surface intersections by ray tracing [Nishita 1990] using Bézier Clipping because calculation of the ray/surface intersection is relatively inexpensive.

3 BASIC CONCEPT

A parametric surface is expressed in terms of parameters u and v. Line drawings are obtained by displaying visible segments of iso-parametric curves with the interval given by a user. In visibility tests, these iso-parametric curves can be considered to correspond to the edges of a polyhedron, and the visibility of a surface thus determines the visibility of curves on it. Though surfaces of a polygon are classified into front faces and back faces, curved surfaces are classified into front faces, back faces, and faces having both visible and invisible regions. We refer these faces as F-faces, B-faces, and S-faces, respectively. Only S-face may have silhouettes. A silhouette curve generally intersects the boundary of a curved surface and we refer to this intersection point as a silhouette point.

We wish to also handle inter-penetrating curved surfaces. Therefore, in drawing curves, visibility may change at intersections on the projection plane between a curve and the boundary of a curved surface, silhouettes, trimming curves, or at inter-penetration points of a curve and a curved surface in 3D space. Therefore, we will present efficient and robust techniques for extracting silhouette curves, testing for intersection between curves on the projection plane, and testing for intersection between a curved surface and a curve. (Ray/surface intersections can be detected using the technique discussed in [Nishita 1990].)

Closed curved surfaces are assumed. Non-closed surfaces can also be processed by treating back faces as front faces. Processing consists of the following ideas:

(1) Curves such as iso-parametric curves are processed without polygonal approximation. Bézier Clipping is used to extract surfaces which overlap the curve on the projection plane, and to test for intersection between curves.

(2) Surfaces are classified according to direction of the surface normal using a hodograph, and silhouette curves are extracted using a hodograph in combination with Bézier Clipping.

(3) Silhouettes, intersections of pairs of curved surfaces, and contour lines are obtained as Bézier curves in the parametric space, like pattern curves.

(4) Tests for intersections between an iso-parametric curve and trimming curves or silhouettes, all these curves are belonging to one surface, are achieved in parametric space by the Bézier Clipping method.

(5) Even though trimming and pattern curves expressed by Bézier form in parametric space become high order rational curves on the projection plane, these curves are processed precisely as in (1) above.

Silhouette curves are extracted before hidden line elimination. The outline of the procedure is as follows:

(1) Surface control points are perspective transformed.

(2) Surfaces are classified into F-face, B-face, or S-face.

(3) Silhouette curves are extracted.

(4) Curved surfaces overlapping with the current curve are determined.

(5) Intersections between curved surfaces and the curve are calculated, and visible parts of the curve are extracted.

4 INTERSECTION TEST ON PROJECTION PLANE

Our intersection test is performed on projection plane. Therefore, we will describe perspective projection, processing to extract surfaces needed for the intersection test, and the intersection test using the Fat Line Method.

4.1 Perspective Transformation of Bézier Surfaces

Let's assume that a Bézier surface of degree n, with control points $\hat{\mathrm{P}}_{ij}(\hat{X}_{ij}, \hat{Y}_{ij}, \hat{Z}_{ij})$ defined in the eye coordinate system. Then the point $\hat{\mathrm{P}}(X(u,v), Y(u,v), Z(u,v))$ on a surface is expressed by the following equation;

$$\hat{\mathrm{P}}(u,v) = \frac{\sum_{i=0}^{n}\sum_{j=0}^{n} W_{ij}\hat{P}_{ij}B_i^n(u)B_j^n(v)}{\sum_{i=0}^{n}\sum_{j=0}^{n} W_{ij}B_i^n(u)B_j^n(v)} \tag{1}$$

where W_{ij} is the weight of a control point, and B is the Bernstein polynomial, given by $B_i^n(u) = \binom{n}{i}u^i(1-u)^{n-i}$. Now, assume the projection plane is $Z = R$, where R is the distance between the projection plane and the viewpoint. Curved surfaces on the projection plane are rational

Bézier surfaces as expressed in the following pair of equations (small letters are used to express perspective-transformed coordinates);

$$x(u,v) = \frac{\sum_{i=0}^{n}\sum_{j=0}^{n} w_{ij}\, x_{ij}\, B_i^n(u) B_j^n(v)}{\sum_{i=0}^{n}\sum_{j=0}^{n} w_{ij}\, B_i^n(u) B_j^n(v)},$$
$$y(u,v) = \frac{\sum_{i=0}^{n}\sum_{j=0}^{n} w_{ij}\, y_{ij}\, B_i^n(u) B_j^n(v)}{\sum_{i=0}^{n}\sum_{j=0}^{n} w_{ij}\, B_i^n(u) B_j^n(v)}. \tag{2}$$

The homogeneous coordinates (x_{ij}, y_{ij}, w_{ij}) of each control point P_{ij} of a surface on the projection plane are given by

$$x_{ij} = R\hat{X}_{ij}/\hat{Z}_{ij}, \quad y_{ij} = R\hat{Y}_{ij}/\hat{Z}_{ij}, \quad w_{ij} = W_{ij}\hat{Z}_{ij} \tag{3}$$

where w_{ij} is a control point weight. If a surface is nonrational (i.e., $W_{ij} = 1$), w_{ij} represents the depth from the view point.

4.2 Classification of Surfaces

After perspective transformation, curved surfaces are classified into F-faces, B-faces, and S-faces which generally include silhouette curves. The z component of the normal is positive at every point of a front face, and negative at every point of a back face.

Surface normals are given by the vector product of tangent vectors P_u and P_v of the curved surface $P(u,v)$ (i.e., $P_u = \partial P(u,v)/\partial u$, $P_v = \partial P(u,v)/\partial v$). The z component of the normal is expressed by

$$N_z = x_u y_v - x_v y_u. \tag{4}$$

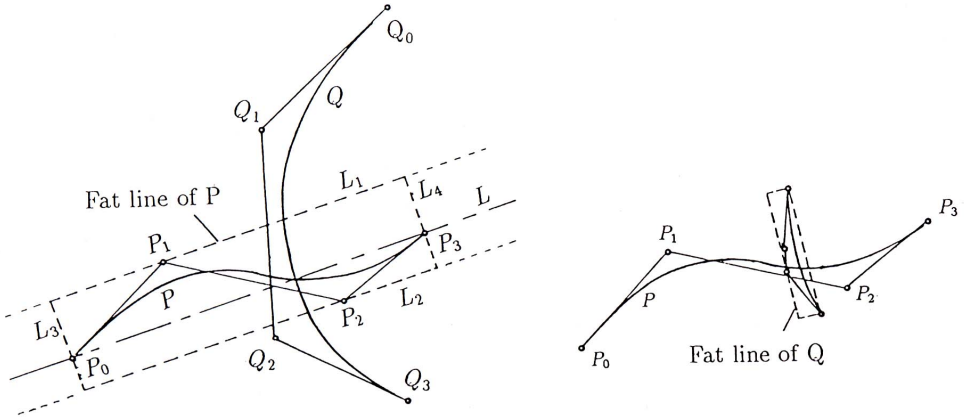
Therefore, the classification can be performed using only (x_u, y_u) and (x_v, y_v), the x and y components of P_u and P_v. The sign of the z component of the normal is determined geometrically by using tangent vectors P_u and P_v of the curved surfaces (see Appendix A).

4.3 Intersections of Curves with Curves

As described in the previous subsection, because the projection of a curved surface onto a plane is a rational Bézier surface, the projected boundary curves and iso-parametric curves are also rational Bézier curves. Therefore, we will now address tests for intersection between rational Bézier curves. The *Fat Line Method* provides robust convergence of iteration to the intersection point of high order rational Bézier curves by using the convex hull property [Sederberg 1990a]. We have modified this method as noted in Appendix B. Since a fat line is an inclined bounding box, curves with no intersection points are rapidly found by this method.

Consider the curves P and Q shown in Fig. 1. First, curve P's fat line is computed as discussed in Appendix B. The fat line is a rectangle which bounds all control points of P, and the longer sides of the rectangle are parallel to the line connecting the endpoints of P. In other words, a curve is viewed as a line with finite width.

The long sides of P's fat line are used to extract an interval of curve Q in parametric space which overlaps P, and the remaining parts are discarded. The extracted interval of Q is of course shorter and straighter than the original Q (Fig. 1(b)). Next the fat line of the shortened Q is computed and similarly used to extract an interval of P overlapping the shortened Q. As this procedure iterates, the width of fat line becomes arbitrarily small, and the overlapped region extracted converges to the intersection point of the curves.



(a) Clipping Q with the fat line of P. (b) Clipping P with the fat line of Q.

Figure 1: Calculation of the intersection point by the Fat Line method.

5 EXTRACTION OF CURVED SURFACES OVERLAPPING WITH CURVE

In order to test for intersecting curves, we need to be able to extract projections of curved surfaces overlapping the projection of the curve being processed. The surfaces are extracted using fat lines. If all control points of a surface lie outside the curve's fat line, the surface does not overlap with the curve. Consider lines L_1, L_2,L_3,and L_4 in Fig. 1. If all the control points of a surface lie to the left of L_1(the direction of P_0 to P_3 is considered) or on the right side of L_2, the surface does not overlap with the curve. Otherwise, the surface may intersect L_1 and L_2.

After clipping the surface by L_1 and L_2, the process is repeated, clipping the surface by L_3 and L_4. Fig. 2 shows the subpatch generated after clipping by L_1 and L_2 (dotted line), then by L_3 and L_4. The method used to clip a curved surface by straight lines is described in [Nishita 1990]. When a curved surface and a curve overlap on the projection plane, if the curved surface lies behind the curve the intersection test is unnecessary. This depth relationship is determined by comparing the maximum and minimum depth of control points of the surface and the curve.

When the surface and the curve overlap, it is necessary not only to test for intersection of the curve and the boundary of the surface, but also to compute silhouette curve intersection points. If the depths of the curve and the curved surface overlap, it is further necessary to calculate the penetration point. The depth and penetration tests, and extraction of silhouette curves, are performed efficiently using subpatches. A patch is repeatedly subdivided until the subpatches provide a good test criterion. For surfaces which overlap the fat line, the following procedure is performed:

1) Subpatches overlapping the fat line are determined.

2) If the subpatch is B-face or lies behind the curve, it cannot hide the curve, and the intersection test is unnecessary.

3) The curve is examined for intersection with the edge of the subpatch which is also an edge of the original surface (edge Q_0Q_3, drawn with a thick line in Fig. 2).

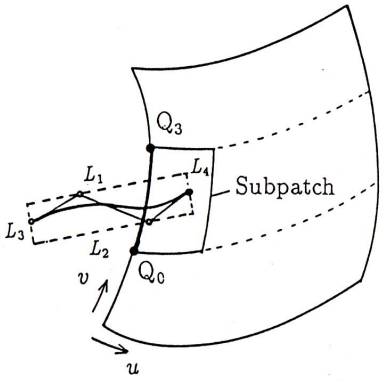


Figure 2: A surface overlapping a curve.

4) If the subpatch is S-face, the intersection point of the curve and the subpatch's silhouette curve is calculated.

5) If the curve does not lie in front of the subpatch, a penetration test is applied. Penetration is determined using a *Fat Face* method analogous to the Fat Line method and discussed in the next section. The fat face bounds the surface's control points.

6) After running steps 3), 4) and 5), if no intersection point between the curve and the surface has been detected, a single arbitrary point (e.g., P_1 in Fig. 13) on the curve is examined to determine whether it is hidden by the surface. If the point is hidden, the whole interval of the curve must also be hidden by the surface. The visibility test is performed by ray tracing using Bézier Clipping [Nishita 1990].

When the surface is subdivided in steps 4) and 5), clipping by the subpatch yields a curve closer to a straight line as noted in Appendix C. Thus, the curve is shortened and the width of fat line is reduced, increasing the efficiency of the curve intersection test and the penetration test.

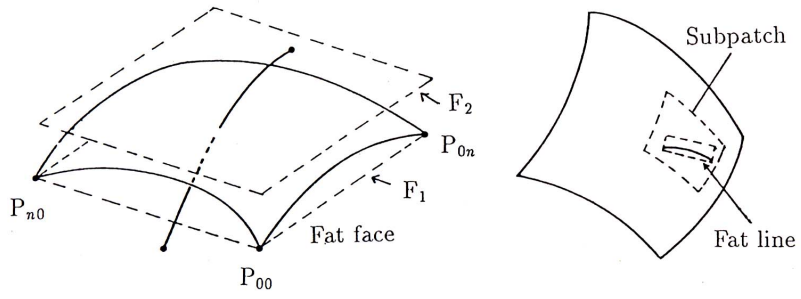
After all intersection points between the curve and surfaces are found, quantitative invisibilities [Appel 1967] are calculated. Using quantitative invisibility, visible segments are easily extracted(see Appendix D).

The segments of the curve with quantitative invisibility zero are visible, and drawn with high precision by the following method. That is, a segment of the curve is subdivided recursively until necessary flatness is obtained. The flatness test uses the same idea, calculating the width of the fat line as mentioned in section 4.3.

6 DISPLAYING PENETRATION CURVES

6.1 Intersection Point between Curve and Surface

As shown in Fig. 3a, the fat face is defined as a region surrounded by a pair of planes F_1 and F_2 which are parallel to the plane constructed by three of the surface's corner points (for example, P_{00}, P_{n0}, and P_{0n}). That is, a curved surface is regarded as a plane with thickness. Space is divided



(a) Clipping curve by fat face in 3D space.

(b) Clipping surface by fat line on projection plane.

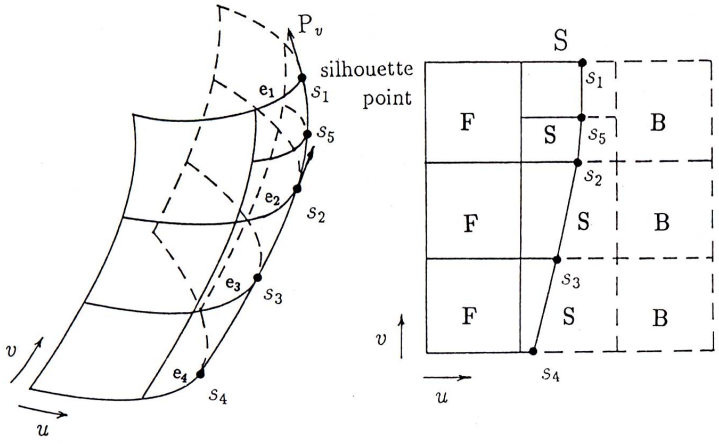
Figure 3: Calculation of the penetration point.

into three regions by F_1 and F_2. When any of the control points is included in the subspace between F_1 and F_2, the penetration test is performed. Specifically, when F_1 lies behind F_2, if all the curve's control points are located in front of F_2 (the subspace including the viewpoint), the curve lies in front of the curved surface. When all the control points are located behind F_1, the curve lies behind the surface. Otherwise, the penetration test must be performed. First, the curve is clipped by the surface's fat face, yielding a shorter curve (see Fig. 3a). Then the surface is clipped by the shortened curve's fat line (Fig. 3b). Iterating this procedure, both the surface and the curve converge to the intersection point.

6.2 Intersection Curve between two Curved Surfaces

The processing of penetrated curved surfaces often occurs in CAGD. One advantage of our method is that it can handle penetrated surfaces. As the degree of the intersection curve between curved surfaces is very high, not only calculation of the intersection curve but also hidden line removal are important. For example, the intersection curve between bicubic surfaces is of degree 324 [Sederberg 1991]. Obviously, approximation is necessary to treat a curve of such high degree. Two types of technique are commonly used to calculate surface/surface intersection curves. The marching method traces the curve starting from an intersection point, such as that between a surface edge and another surface. Subdivision methods divide the surface into subpatches, approximates the subpatches as planes, and calculates the intersection lines between the planes [Bhat 1990]. Both of these methods represent intersection curves in terms of a sequence of straight lines connecting intersection points, and therefore lack precision and smoothness.

Therefore we use a method of calculating the intersection curve directly [Sederberg 1991]. In this method, a single intersection curve between curved surfaces is defined in the parameter spaces of both curved surfaces. That is, the intersection curve is dealt with two different mathematical equations in the parameter spaces. Two equations of the intersection curve, one on each surface, are to be determined. Now we address the processing procedure. First, the endpoints of the intersection curve are calculated as described above (Section 6.1). At two endpoints these two curves can be connected with C^k continuity. Here, we will consider C^1 continuity. Each obtained curve is a Bézier curve with degree three in parametric space for a bicubic surface. The control points of the two intersection curves obtained are compared, and subdivision continues until the



(a) Projection plane.

(b) uv space.
F : F-face
B : B-face

Figure 4: Extraction of a silhouette curve.

necessary accuracy is obtained. Each intersection curve is mapped onto its surface and projected onto the projection plane, then the curve is expressed by a rational Bézier curve with degree 18. The fat line method is useful for a curve of such high degree.

7 EXTRACTION OF SILHOUETTE CURVES

Our method can extract silhouette curves with a high degree of accuracy. Extraction of shilouette curves is performed only on S-faces. The S-face is divided into $N \times N$ elements (in this paper subdivision is performed with the interval of iso-parametric curves for display). After subdivision on each element silhouette curves are extracted as follows(see Fig. 4);

1) The elements are classified as F, B, or S-faces.

2) If the element is S-face, go to next step. Otherwise the next element is processed.

3) Some edges(e_2 and e_3 in Fig. 4) of S-face and boundaries of the surface(e_1 and e_4) have silhouette points(s_1,s_2,s_3 and s_4). These silhouette points are extracted by the method described in section 7.1.

3.1) If silhouette points can not be found, then go to step 4. Otherwise go to step 3.2.

3.2) If two silhouette points are found, go to step 3.3. Otherwise go to step 4.

3.3) The silhouette curve is obtained by linear interpolation between the two silhouette points in the parameter space.

3.4) If the tangent vector directions at the two silhouette points exceeds the given value, then go to step 4. Otherwise go to step 3.5.

3.5) If the z-component of the normal at the midpoint exceeds the specified tolerance, go to step 4. Otherwise the next element is processed.

4) The element is subdivided into two elements, then return to step 1.

The proposed method mentioned above interpolates linearly between two silhouette points in the parameter space. Then in most cases maximum error occurs at the midpoint between silhouette points. Therefore the error at the midpoint is measured and adaptive subdivision is repeated until necessary accuracy is obtained. Thus high precision silhouette curves are extracted.

By the way, when a silhouette curve lies within a curved surface but does not intersect the boundaries of the elements, it must be a closed loop. In this case, Whitted's method can not detect a silhouette curve, because his method searches silhouette points only on the boundaries of a surface.

On the other hand, when no silhouette point is extracted from S-face, the proposed method subdivides the element by step 3.1 and step 4 of the algorithm mentioned above and a single closed loop is found. Furthermore, when more than one silhouette curve is in an element each silhouette curve is extracted by performing subdivision in step 3.2 and step 4. However when both a silhouette curve passing through the edges and a closed silhouette loop exist in an element, the closed loop can not be detected by the proposed method.

7.1 Extraction of Silhouette Points

P_u and P_v are defined as the u and v components of the tangent vector of a surface at an arbitrary point, so the surface normal is $\mathrm{P}_u \times \mathrm{P}_v$. After perspective transformation, the normal at a silhouette point has a 0 z-component. Algebraic expression of this condition yields a polynomial of high degree (e.g., a bicubic surface leads to a polynomial of degree seven). Therefore, it is difficult to obtain the silhouette point analytically. Our method extracts the silhouette curve using the range of x and y components of control points of P_u and P_v after perspective transformation. In essence, the silhouette curve is extracted geometrically using the hodograph in the xy plane.

Consider extraction of a silhouette curve on the surface edge $v = 0$. The edge is expressed by a rational Bézier polynomial and is a function of u only. Using equation(3) to express the control point coordinates P_{ij} on the projection plane, after perspective transformation P_u and P_v are given by

$$\begin{aligned} \mathrm{P}_u(u) &= \frac{\sum_{k=0}^{2n-2} \mathrm{R}_k B_k^{2n-2}(u)}{(\sum_{k=0}^{2n-2} w_{k0} B_k^{2n-2}(u))^2}, \\ \mathrm{P}_v(u) &= \frac{\sum_{k=0}^{n} (\frac{w_{k1}}{w_{k0}}) \mathrm{H}_{k0}^{v} B_k^{n}(u)}{(\sum_{k=0}^{n} w_{k0} B_k^{n}(u))^2} \end{aligned} \tag{5}$$

where H expresses the control points of the hodograph given by equation (12). For example, when $n = 3$, R_k (reference[Sederberg 1987]) is given by

$$\begin{aligned} \mathrm{R}_0 &= 3\mathrm{D}_{01},\ \mathrm{R}_1 = 3/2\mathrm{D}_{02}, \\ \mathrm{R}_2 &= 1/2\mathrm{D}_{03} + 3/2\mathrm{D}_{12}, \\ \mathrm{R}_3 &= 3/2\mathrm{D}_{13},\ \mathrm{R}_4 = 3\mathrm{D}_{23} \end{aligned}$$

$$\mathrm{D}_{ij} = w_{i0} w_{j0} (\mathrm{P}_{i0} - \mathrm{P}_{j0}) = W_{i0} W_{j0} (\hat{Z}_{j0} \hat{X}_{i0} - \hat{Z}_{i0} \hat{X}_{j0}). \tag{6}$$

As mentioned above, the range of the hodograph (the range of slope) of a rational Bézier curve is contained in that of the hodograph when weight w is set to 1. Accordingly, when a subpatch is

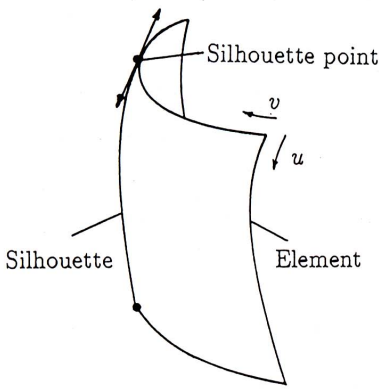


Figure 5: Extraction of a silhouette point.

small and all weights of control points are nearly equal, P_u and P_v on the projection plane can be approximated as follows:

$$\mathrm{P}_u(u) = \sum_{i=0}^{n-1} \mathrm{H}_{i0}^u B_i^{n-1}(u), \quad \mathrm{P}_v(u) = \sum_{i=0}^{n} \mathrm{H}_{i0}^v B_i^n(u). \tag{7}$$

$\mathrm{P}_u \times \mathrm{P}_v = 0$ is satisfied when P_u and P_v are colinear in the projection plane (Fig. 5) (indicating that the slope of P_u and P_v are equal), or when $\mathrm{P}_u = 0$ or $\mathrm{P}_v = 0$ (a silhouette has a cusp).

The range of the slope of P_u is derived from locations of the control points of the hodograph. In other words, it is given by the wedge area determined by lines L_1 and L_2 (Fig. 6). Lines L_1 and L_2 are expressed by

$$\begin{aligned} f^1(x,y) &= a_1 x + b_1 y = 0 \\ f^2(x,y) &= a_2 x + b_2 y = 0. \end{aligned} \tag{8}$$

Lines L_1 and L_2 divide the xy plane into four parts. If P_v lies in region A or B in Fig. 6, the slope of P_v may coincide with that of P_v. If P_v lies outside regions A and B, there is no silhouette curve. The following inequalities determine regions A and B.

$$\begin{aligned} & f^1(x,y) \geq 0 \quad , \qquad f^2(x,y) \leq 0 \\ & \qquad \text{or} \\ & f^1(x,y) \leq 0 \quad , \qquad f^2(x,y) \geq 0 \end{aligned} \tag{9}$$

The interval of P_u where the above inequalities are satisfied gives a silhouette curve. For example, substituting x and y of equation (7) into (8), the upper left inequality of (9) is expressed by

$$\begin{aligned} f^1(u) &= \sum_{i=0}^{n} (a_1 \mathrm{x} + b_1 \mathrm{y}) \mathrm{B}_i^n(u) = \sum_{i=0}^{n} f_{i0}^1 \mathrm{B}_i^n(u) > 0 \\ f_{i0}^1 &= a_1 \mathrm{x}_{i0}^u + b_1 \mathrm{y}_{i0}^u. \end{aligned} \tag{10}$$

Thus the interval of u satisfying this inequalities can be obtained by Bézier Clipping. Intervals of u satisfying the remaining inequalities of (9) are calculated in a similar manner. Next, the

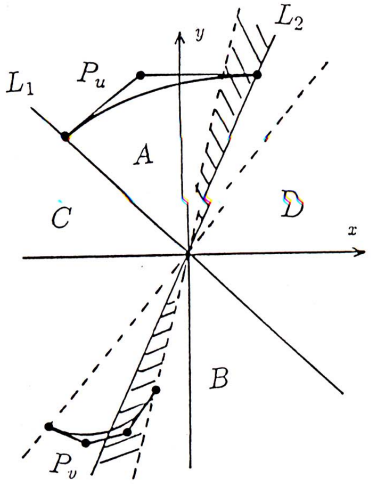


Figure 6: Extraction of silhouette point using the hodograph.

wedge of P_u is calculated, and the interval of the solution of P_v is obtained. The solution lies in the intersection of the intervals of u calculated from P_u and P_v. Therefore the parts outside the intersection of these intervals are clipped away, and the procedure is repeated. Thus the silhouette point's u can be obtained with the necessary accuracy by iteration. If the interval of u is not reduced by iteration, more than one silhouette point exists on the boundary, so the surface is subdivided and the procedure is repeated, guaranteeing detection of all silhouette points. w_{ij} are functions of parameters u and v. As subdivision is repeated, each parameter range becomes narrow. Therefore all values of w_{ij} come close to the same value. Thus the difference in weight of the control points is reduced, and precise silhouette points are obtained, even with approximated inequalities used for (9).

8 DISPLAY OF TRIMMING AND PATTERN CURVES ON SURFACES

When curves are mapped onto approximated polygons, a number of difficulties arise. First, it is difficult to define a curve spanning more than one polygon. Moreover, because the tangent is discontinuous at the boundary of polygons, the curves are also discontinuous. Therefore, in displaying curves on curved surfaces, it is very useful to avoid polygonal approximation.

The trimmed patches have come into common use in CAGD. In this paper a curve representing a trimming curve or a pattern on a surface (called a pattern curve) is defined in uv space and expressed by a Bézier curve of degree n. Even though the expression of trimming curves is the same as that of pattern curves, the trimming curve affects to the visibility of the other curves when the curve intersect with the trimming curve.

There are two types of the intersection test between iso-parametric curves and trimming curves. The test for an iso-parametric curve and a trimming curve on one surface is performed in parametric space. Since this test is performed between the straight line (i.e., iso-parametric curve) and the Bézier curve(generally degree 3), the test is simple, and the fat line method is employed in parametric space. The intersection test between an iso-parametric curve and a trimming curve belonging to different surfaces each other is performed on the projection plane. Equation (1)

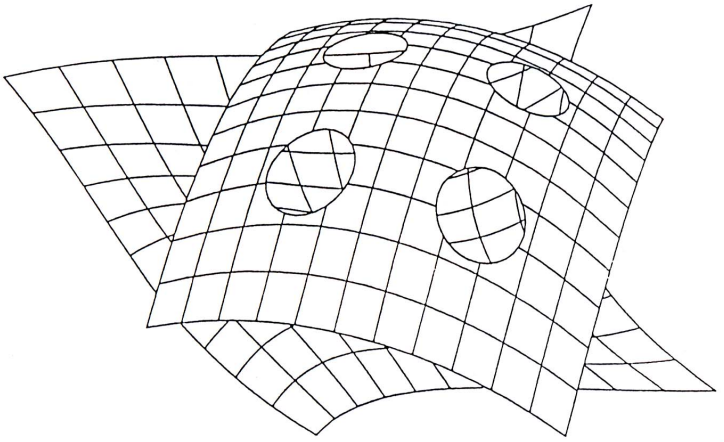


Figure 7: A calculated example of trimmed patches.

yields the coordinates of the curve on the projection plane. For example, a bicubic surface and a cubic pattern curve leads to a Bézier curve of degree 18 in 3D space. After transformation, hidden line removal is performed in the same way as for iso-parametric curves. Whether a pattern curve lies on a front or a back face can be determined by using the uv bounding box of the curve to divide the surface, and examining the orientation of the resulting surface. Fig. 7 shows the calculated example of trimmed surfaces.

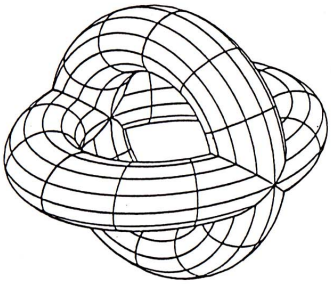
9 CONTOUR LINES

Contour lines are useful to convey the height relative to the horizontal plane, the distance from the viewpoint, and smoothness of the curved surface. Each contour line is determined by the intersection of a plane at a given slope (often horizontal), and the curved surface. Contour lines can be extracted using the technique developed for the penetration test, which determined intersections between curves and surfaces. The surface is divided into $N \times N$ elements and the element edges are examined for intersections with the plane (as we did for silhouette curves). Contour lines are then obtained by Hermite interpolation in uv space between pairs of intersection points [Sederberg 1991].

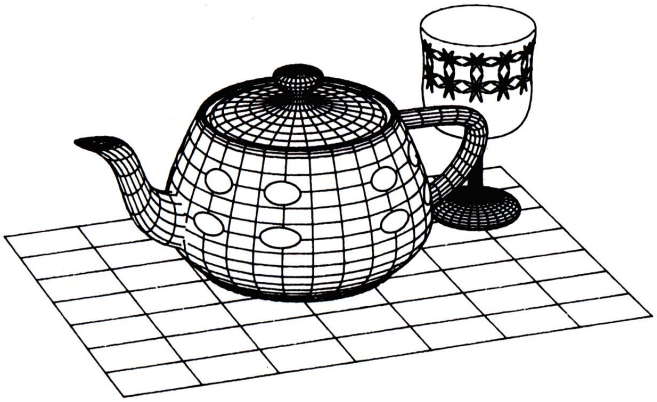
If multiple intersection points are detected, the subpatch is further subdivided. The intersection test is performed using Bézier Clipping. As the obtained contour lines are expressed in uv space, they can be treated as curves on surfaces, and hidden line removal can be performed as for pattern curves.

10 EXAMPLES

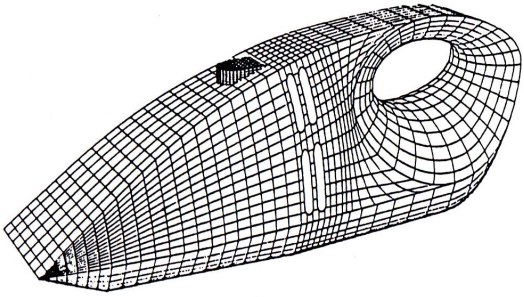
Four examples are shown in Fig. 8. (a) shows an example of inter-penetrating tori. Figure (b) shows an example of pattern curves affixed to the Utah teapot. (c) shows a commercial product design (a hand-held vacuum cleaner). (d) shows contour line generated by intersections of the teapot surface with tilted parallel planes. The drawings were calculated in 24.2 seconds (a), 72.0 seconds (b), and 25.3 seconds (c), 85.0 seconds (d) on a Silicon Graphics IRIS 4D/25G workstation. In these examples, the average number of iterations for curve/curve intersection on the projection



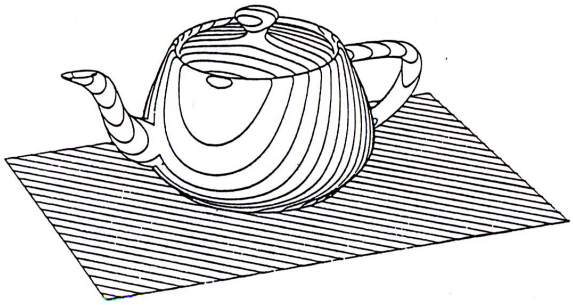
(a) Torus



(b) Teapot



(c) Vacuum cleaner



(d) Contour lines

Figure 8: Examples.

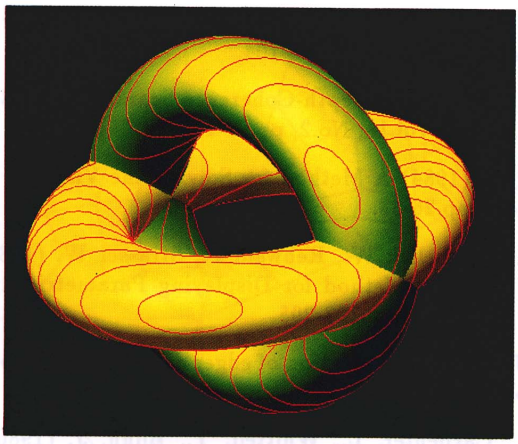


Figure 9: Shaded image composited with contour lines

plane, with tolerance in the parameter domain of 10^{-3}, are as follows: Pairs of rational curves of degree 3: 1.6 iterations. Intersecting rational curves of degree 3 with degree 6 curves: 1.7 iterations. Intersecting rational curves of degree 3 with degree 18 curves: 2.7 iterations. These results illustrate the efficiency of our intersection test for rational curves of high degree.

Fig. 9 shows an image in which depth contour lines have been composited with a shaded image. The shaded image was displayed with a scan line algorithm [Nishita 1991].

11 CONCLUSION

The intersections between curves, classification of surfaces as front or back, extraction of silhouette curves, calculation of penetration points, and extraction of contour lines, all important operations for hidden line elimination, are performed by Bézier Clipping. The method has the following advantages:

1) By avoiding polygonal approximation, high precision display is obtained, with smaller memory requirement.

2) High precision display is obtained even for inter-penetrating surfaces.

3) Figure curves on surfaces and contour lines can be drawn with high precision.

4) Silhouette curves can be extracted with high precision.

REFERENCES

[Appel 1967] Appel A.(1967) The Notion of Quantitative Invisibility and the Machine Rendering of Solids. *Proc. 22nd National Conference ACM* , pp. 387-393.

[Bhat 1990] Bhat S.(1990) Bézier Surface/Surface Intersection. *IEEE CG & A*, Vol.10, pp.50-58.

[Elber 1990] Elber G., Cohen E.(1990) Hidden Curve Removal for Free Surfaces. *Computer Graphics*, Vol.24, No.4, pp.95-104.

[Griffths 1984] Griffths JG.(1984) A Depth-Coherence Scanline Algorithm for Displaying Curved Surfaces. *CAD*, Vol.16, No.2, pp.91-101.

[Hornung 1984] Hornung C.(1984) A Method for Solving the Visibility Problem. *IEEE CG & A*, Vol.4, No.7, pp.26-33.

[Hornung 1985] Hornung C., Lellek W., Rehwald P., Strasser W. (1985) An Area Oriented Analytical Visibility Method for Displaying Parametrically Defined Tensor-Product Surface. *CAGD*, 2, pp.197-205.

[Kripac 1985] Kripac J.(1985) Classification of edges and its application in determining visibility, *CAD*,Vol.17, No.1, pp.30-36.

[Lane 1980] Lane, J., Carpenter, L., Whitted, T., Blinn, J. (1980) Scan Line Methods for Displaying Parametrically Defined Surfaces. *CACM*, Vol.23, No.1, pp.23-34.

[Li 1988] Li L.(1988) Hidden-Line Algorithm for Curved Surfaces. *CAD*, Vol.20, No.8, pp.466-470.

[Loutrel 1970] Loutrel P.(1970) A Solution to the Hidden-Line Problem for Computer-Drawn Polyhedra. *IEEE Transactions on Computers*, Vol.19, No.3, pp.205-213

[Nishita 1990] Nishita, T., Sederberg, T.W., Kakimoto, M. (1990) Ray Tracing Rational Trimmed Surface Patches. *Computer Graphics*, Vol.24, No.4, pp.337-345.

[Nishita 1991] Nishita, T., Kaneda, K., Nakamae, E.(1991) Scanline algorithm for displaying trimmed surfaces by using Bézier clipping. *The Visual Computer*, Vol.7, No.5-6, pp.269-279.

[Ohno 1983] Ohno Y.(1983) A Hidden Line Elimination Method for Curved Surfaces. *CAD*, Vol.15, No.4, pp.209-216.

[Pueyo 1987] Pueyo, X., Brunet, P.(1987) A Parametric-Space-Based Scan-Line Algorithm for Rendering Bicubic Surfaces. *IEEE CG & A*, Vol.7, No.8, pp.17-24.

[Schweitzer 1982] Schweitzer, D., Cobb E. S.(1982) Scanline Rendering of Parametric Surfaces. *Computer Graphics*, Vol.16, No.3, pp.265-271.

[Sederberg 1987] Sederberg, T.W., Wang, X.(1987) Rational Hodographs. *CAGD*, Vol.4, pp.333-335.

[Sederberg 1990a] Sederberg, T.W., Nishita, T.(1990) Curve Intersection using Bézier Clipping. *CAD*, Vol.22, No.9, pp.538-549.

[Sederberg 1990b] Sederberg, T.W., Meyers(1990) Loop Detection in Surface Patch Intersections. *CAGD*, Vol.5, pp.161-171.

[Sederberg 1991] Sederberg, T.W., Nishita, T.(1991) Geometric Hermite Approximation of Surface Patch Intersection. *CAGD*, Vol.8, pp.97-114.

[Sequin 1985] Sequin C., Wisney P.(1985) Visible Feature Return at Object Resolution. *IEEE CG & A*, Vol.5, No.5, pp.27-50.

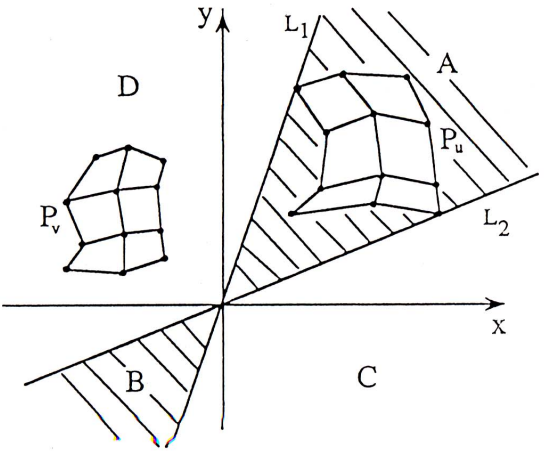


Figure 10: A front/back test of a surface.

Appendix

A Classification of Surfaces

Surface normal is expressed in equation (4). We examine the sign of N_z geometrically by using a hodograph. This hodograph is defined as the first derivative and its graph is shown in Fig. 10. Therefore, the classification can be performed using only (x_u, y_u) and (x_v, y_v), the x and y components of P_u and P_v. P_u and P_v are rational Bézier surfaces, and it is known that the range of their tangent direction is bounded by the hodograph of nonrational (i.e., $w = 1$) surfaces [Sederberg 1987]. If surfaces are nonrational, P_u and P_v are expressed by

$$\begin{aligned} \mathrm{P}_u(u,v) &= \sum_{i=0}^{n-1}\sum_{j=0}^{n} \mathrm{H}^u_{ij} B^{n-1}_i(u) B^n_j(v), \\ \mathrm{P}_v(u,v) &= \sum_{i=0}^{n}\sum_{j=0}^{n-1} \mathrm{H}^v_{ij} B^n_i(u) B^{n-1}_j(v) \end{aligned} \tag{11}$$

where H^u_{ij} and H^v_{ij} are derived from the projected control points (see Eq.(3)) and given by

$$\mathrm{H}^u_{ij} = n(\mathrm{P}_{i+1,j} - \mathrm{P}_{i,j}), \quad \mathrm{H}^v_{ij} = n(\mathrm{P}_{i,j+1} - \mathrm{P}_{i,j}). \tag{12}$$

As shown in Fig. 10, the xy plane is divided into four regions A, B, C, and D by lines L_1 and L_2, which bound the wedges which express the range of the slope of the projection of P_u on the xy plane. If P_v is entirely contained in region D, the surface is an F-face. If it is entirely contained in region C, the surface is a B-face. If a part of P_v is contained in region A or B, the surface is an S-face. The region containing P_v is determined by the function described in Appendix B(see equation (13)).

B Intersection Test of Rational Bézier Curves by the Fat Line Method

Consider two rational Bézier curves, P(s) of degree n with control points P_i, and Q(t) of degree m with control points Q_j. The fat line of a curve segment is the rectangle bounded by the lines L_1,

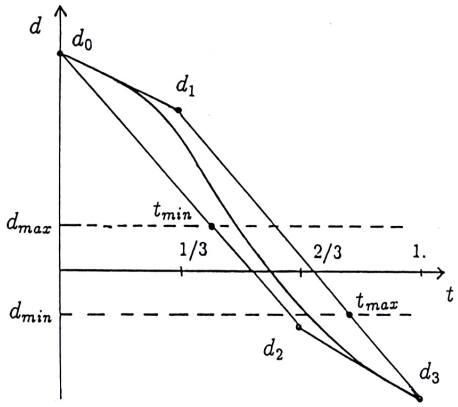


Figure 11: Distance function.

L_2, L_3, and L_4, shown in Fig. 1. The distance $d(x, y)$ from an arbitrary point (x, y) to the line L which passes through endpoints P_0 and P_n of P(s) is given by

$$d(x, y) = ax + by + c \tag{13}$$

where (a, b) is the unit normal of L (i.e., $a^2 + b^2 = 1$). Within the region enclosed by L_1 and L_2, which are edges of the fat line, the following relation is satisfied:

$$d_{max} \geq d(x, y) \geq d_{min} \tag{14}$$

where

$$d_{max} = \max\{d_0, d_1, ..., d_n\} \quad , \quad d_{min} = \min\{d_0, d_1, ..., d_n\} \tag{15}$$
$$d_i = d(x_i, y_i)$$

and d_i is the distance from control point P_i to line L.

On the other hand, curve Q(t) is expressed by

$$Q(t) = \frac{\sum_{j=0}^{m} w_j Q_j B_j^m(t)}{\sum_{j=0}^{m} w_j B_j^m(t)}. \tag{16}$$

Substituting x and y of Q(t) into (13) yields the following equation, a function of t only.

$$d(t) = \frac{\sum_{j=0}^{m} w_j d_j B_j^m(t)}{\sum_{j=0}^{m} w_j B_j^m(t)}. \tag{17}$$

Therefore, it is possible to find the interval of t satisfying the relation $d_{max} \geq d(t) \geq d_{min}$ using Bézier Clipping. The distance function d is a non-parametric rational Bézier curve and the coordinates of its control points are $(j/n, d_j)$. Fig. 11 shows the distance function corresponding to the cubic Bézier curve of Fig. 1. As shown in the figure, interval $[t_{min}, t_{max}]$ which satisfies the above condition is found using the convex hull property of d. As there is no solution in intervals

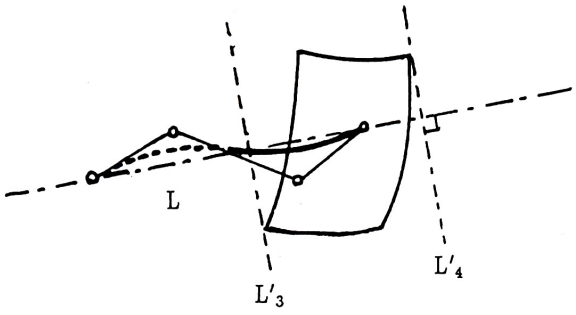


Figure 12: Curve clipping by a curved surface.

other than $[t_{min}, t_{max}]$, intervals $[0, t_{min}]$ and $[t_{max}, 1]$ of Q can be discarded. Next the fat line of the clipped curve Q is calculated and used to clip curve P. Iterating this procedure converges on the intersection point.

Even when multiple intersection points exist, this method finds the solution robustly. The method is discussed greater detail in reference [Sederberg 1990a]. This method is an extension of [Sederberg 1990a] because in [Sederberg 1990a] the following two functions are used instead of equation (17).

$$d^1(t) = \sum_{i=0}^{n} d_i^1 B_i^n(t) \geq 0, \quad d^2(t) = \sum_{i=0}^{n} d_i^2 B_i^n(t) \leq 0 \tag{18}$$

where

$$\begin{aligned} d_i^k &= w_i(ax_i + by_i + c^k) \quad (k = 1, 2), \\ c^1 &= c - d_{min}, \; c^2 = c - d_{max}. \end{aligned} \tag{19}$$

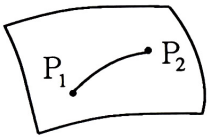
In the proposed method, only the coordinates of the control points $(j/n, d_i)$ of the rational curve, d, are used, while the method in [Sederberg 1990a] requires the calculation of equation (19) for getting the the control points of two nonrational curves, d^k.

C Clipping a Curve by a Surface

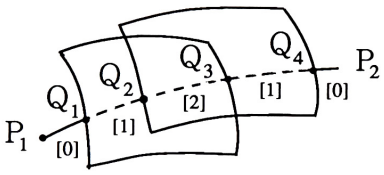
Testing for intersection between a curve and the edge of a surface, as well as calculation of a penetration point between a curve and a surface are performed by iterated Bézier Clipping. Because the efficiency of the method is greater for straighter curve segments, it is better to clip the curve before performing these tests. As shown in Fig. 12, line L passes through the endpoints of the curve. Lines L'_3 and L'_4, which are orthogonal to the line L, are calculated from interval $[d_{min}, d_{max}]$ of control points of the curve defined in equation (16). Clipping is achieved as described in Appendix B using lines L'_3 and L'_4.

D Visibility Test

As is well known, quantitative invisibility is the number of front surfaces which hide a curve when viewed from a given viewpoint. Let's explain the visibility test using Fig. 13.



(a) Quantitative invisibility in case of having no intersection



(b) Quantitative invisibility in case of having intersections

Figure 13: Visibility test for a curve.

As shown in Figure (a), if there is no intersection point between the curve and the surfaces, an arbitrary point on the curve is examined to see if the point is hidden. If the point is hidden, the quantitative invisibility is 1. Otherwise it is 0. When the curve intersects with the surfaces, Figure (b) illustrates how quantitative invisibility varies on curve P_1P_2. Q_1, Q_2, Q_3 and Q_4 are intersection points between the curve and the surfaces hiding them. First, the quantitative invisibility at starting point P_1 is calculated by using the ray tracing method [Nishita 1990]. In this figure it is 0, because no surface hides the point. Quantitative invisibility is increased by 1 at points Q_1 and Q_2, because the curve goes behind the surfaces. Meanwhile, quantitative invisibility is decreased by 1 at points Q_3 and Q_4, because the curve emerges from the surfaces. Whether the curve goes behind or comes out can be judged by using the vector product of the tangent vectors of both the curve and the boundary of the surface at an intersection point. The segments with a quantitative invisibility of zero are visible.

Tomoyuki Nishita is a professor in the department of Electronic and Electrical Engineering at Fukuyama University, Japan. *He was on the research staff at Mazda from 1973 to 1979 and* worked on design and development of computer-controlled vehicle system. He joined Fukuyama University in 1979. He was an associate researcher in the Engineering Computer Graphics Laboratory at Brigham Young University from 1988 to the end of March, 1989. His research interests involve computer graphics including lighting model, hidden-surface removal, and antialiasing.

Nishita received his BE, ME and Ph. D in Engineering in 1971, 1973, and 1985, respectively, from Hiroshima University. He is a member of ACM, IPS of Japan and IEE of Japan.

Address: Faculty of Engineering, Fukuyama University, Sanzo, Higashimura-cho, Fukuyama, 729-02 Japan.
E-mail: nis@eml.hiroshima-u.ac.jp

Shinichi Takita is a professor in the Department of Education at Kagawa University,Japan. His research interests include computer graphics and CAI.

Takita received his BE and ME degrees in electrical engineering form Hiroshima University in 1964 and 1966, respectively. He is a member of the IEE of Japan and the Japan Society of Industrial and Technical Education.

Address: Faculty of Education, Kagawa University,
1-1, Saiwai-cho, Takamatsu, 760 Japan.

Eihachiro Nakamae is a professor at Hiroshima University where he was appointed as research associate in 1956 and a professor in 1968. He was an associate researcher at Clarkson College of Technology, Potsdam, N. Y., from 1973 to 1974. His research interests include computer graphics and electric machinery.

Nakamae received the BE, ME, and DE degrees in 1954, 1956, and 1967 from Waseda University. He is a member of IEEE, IEE of Japan, IPS of Japan and IEICE of Japan.

Address: Faculty of Engineering, Hiroshima University,
Kagamiyama, Higashi-hiroshima, 724 Japan.
E-mail: naka@eml.hiroshima-u.ac.jp