Electronics and Communications in Japan, Part 3, Vol. 84, No. 3, 2001
Translated from Denshi Joho Tsushin Gakkai Ronbunshi, Vol. J82-A, No. 12, December 1999, pp. 1853–1861

Fast Display of Curves and Surfaces with Correct Topology in All Resolutions

Yoshiyuki Kanazawa

LCD Projector Systems Division, Sharp, Ltd., Yaita, Japan 329-2193

Kenichi Kanatani

Department of Computer Science, Gunma University, Kiryu, Japan 376-8515

SUMMARY

We present an efficient method for displaying curves and surfaces defined by implicit equations in such a way that the topology (connection) of the displayed curve is correct in whatever resolution. This is a consequence of detecting all the critical points of the curve and recursively subdividing the display region. Critical points are detected by evaluating the intervals, mean-value forms, and Krawczyk forms of the functions involved. We apply our method to visualizing surfaces by drawing their occluding contours. © 2000 Scripta Technica, Electron Comm Jpn Pt 3, 84(3): 1–11, 2001

Key words: User interface; interval analysis; critical point; curve; surface; occluding contour.

1. Introduction

Visually displaying curves and surfaces is not only one of the most basic procedures of computer graphics and CAD but also an indispensable tool for education of mathematics. This paper deals with curves and surfaces defined by *implicit equations*, that is, as the set of zeros of functions. Most of the curves and surfaces encountered in textbooks on mathematics are defined by such implicit equations. Displaying curves and surfaces would be much simpler if their coordinates are given as explicit functions of parameters: All we need to do is successively increment the parameters and draw the resulting trajectories. However, curves and surfaces defined by implicit equations are not so easy to display. This paper presents a fast method for this purpose.

Taniguchi and Sugihara [19] presented a method for displaying algebraic curves and surfaces defined by implicit equations given by polynomials in the coordinates, but their method is very inefficient. It is true that curves and surfaces used for CAD are mostly algebraic and the theory of algebra can easily be applied in many useful ways. In this paper, however, we do not limit ourselves to polynomials. We consider curves and surfaces defined by implicit equations that can be *written down*, or *elementary functions* to be precise, including trigonometric functions, inverse trigonometric functions, exponential functions, and logarithmic functions.

Today, many symbolic algebra software tools are available for displaying surfaces, but most of them display wireframes with appropriate surface rendering [5], as is

This work was supported in part by the Ministry of Education, Science, Sports and Culture, Japan under a Grant in Aid for Scientific Research C(2) (No. 11680377).

customary in computer graphics and CAD systems. For human eyes, however, this is not a natural way of seeing surfaces. When children draw mountains, they draw their occluding contours. If told to draw a sphere, one would unconsciously draw a circle; if told to draw cones and cylinders, one would draw straight lines and elliptic arcs to represent their rims and occluding contours. This is because people are able to imagine the surface shape very precisely merely by seeing rims and occluding contours. In fact, a lot of research has examined the role of occluding contours in relation to the psychology of human perception and machine interpretation of images [1–3, 8, 12].

However, the visible occluding contours of a wireframe are not the true occluding contours; they are merely the wires in their neighborhoods. Such apparent occluding contours may approach the true occluding contours as the mesh size of the wireframe decreases, but the 3D shape of a dense wireframe would be very difficult to perceive by the human eye. In this paper, we resolve this difficulty by directly drawing the true occluding contours, which are projections of the space curves in three dimensions, called the *contour generators*, onto the image plane or the *retina* [9]. Hence, displaying surfaces reduces to displaying space curves.

2. Displaying Curves

2.1. Curve tracing

One of the best known methods for displaying a curve is to take an initial point and incrementally trace the curve [6, 19]. The increment can be computed by various means—reducing the problem to numerical integration of ordinary differential equations, repeating Newton iterations, and their various combinations and variations. Whatever method we use, however, the following problems cannot be avoided:

- No general methods exist for computing the initial points.
- If the curve has multiple connected components, isolated points, and small loops, it is difficult to determine if they have exhaustively been detected.
- If two segments of the same curve lie very close to each other, we may jump to a wrong segment in the course of tracing the curve.
- The computation of increments fails at singularities where the curve branches out or crosses itself.

Taniguchi and Sugihara [19] tried to avoid these difficulties by using symbolic computation of resolvants and introducing complicated threshold adjustment procedures. As a result, the computational burden was so heavy that only polynomials of degree 6 or less can be dealt with within a practical time. Also, their method cannot be applied to nonalgebraic curves.

2.2. Space subdivision

A contrasting strategy to curve tracing is to subdivide the display region into small cells, compute the function values at the cell vertices, and generate line segments along which the function is supposedly zero. A typical example of uniform space subdivision is the *zero-crossing*, a well-known method in image processing, for detecting edges by searching for pixels at which the Laplacian takes value zero. Recursive subdivision is used in CAD solid modeling systems for finding the boundary representation of an object defined by quad trees and octrees [14, 15]. Although the above-mentioned difficulties of curve tracing are resolved by this type of space subdivision, the following problems arise:

- The topology (connection) below the cell size, or the *resolution*, is not always correct: Some cross points, isolated points, and small loops may be overlooked.
- It is not easy to connect the generated segments correctly to obtain a global representation.
- A large amount of computation is required for subdividing the entire display region.

In this paper, we point out that while the difficulties of curve tracing are inherent and unavoidable, the difficulties of space subdivision can be overcome. The second issue above may be important for a CAD modeling system, by which we want to design an overall structure of the object. If displaying the curve is the sole purpose, as in textbooks of mathematics, we do not need a global representation. The third issue can be resolved if the region is subdivided coarsely. The biggest obstacle is the first issue. In the following, we present a technique for displaying curves with correct topology however coarse the subdivision is. This is made possible by finding all critical points using interval analysis techniques.

2.3. Finding all critical points

What determines the topology of a curve is its critical points and singularities. A *critical point* of a curve $f(x, y) = 0$ is a point (x, y) such that

$$f_x(x,y) = 0, \qquad f_y(x,y) = 0 \tag{1}$$

It is called a *singularity* if it satisfies $f(x, y) = 0$ in addition. If the curve intersects with itself or branches out, the intersections and branching points are necessarily singularities. If the curve has loops, they have critical points inside. This implies that in order to test if two segments of the same curve intersect or lie closely nearby, we do not need a complicated symbolic procedure like that of Taniguchi and Sugihara [19] or high enough resolution to distinguish crossings from close encounters. All we need is to detect singularities: The curve crosses itself if it has a singularity and it does not if no singularity exists. Similarly, finding isolated points and loops reduces to finding critical points.

Finding all critical points reduces, in turn, to solving simultaneous nonlinear equations inside the display region. This problem has been well studied in relation to electronic circuit analysis. One of the best-known techniques is the *Krawczyk method*, which searches the given region by recursively subdividing it and testing the existence of solutions. The test is conducted by interval analysis techniques involving the mean-value theorem and the fixed-point theorem for contracting mappings [7, 16, 20, 21]. The Krawczyk method is intended for circuit equations with hundreds of thousands of variables. Since a plane curve has only two variables and a space curve has only three variables, the computation is very efficient in our application.

Our method begins with finding all critical points by the Krawczyk method prior to displaying the curve itself. Then, we subdivide the display region into cells in such a way that the computed critical points are at the cell vertices. Since it is guaranteed that no critical points exist inside the cells, it is also guaranteed that the curve has no cross points or branching points inside any cell. This fact allows us to generate a chain of line segments with correct topology. If smoothness is required, we only need to refine the cell division as finely as necessary.

3. Interval Analysis

3.1. Interval algebra

Interval analysis evaluates the range of a function for given intervals of the input variables [13, 16]. Arithmetic operations for intervals I and I' are defined by

$$I \circ I' = \{xx' \mid x \in I,\, x' \in I'\} \tag{2}$$

where $\circ$ stands for +, –, ×, or /. The actual computation is done by evaluating the maximum and the minimum of the four values resulting from application of the operation $\circ$ to the endpoints of the intervals I and I'. The operation $I \circ I'$ defines a unique interval if the number system is extended by adding ∞ and $-\infty$ to the real numbers. The arithmetic interval operations can easily be extended to general operations including unary operations, trigonometric functions, inverse trigonometric functions, exponential functions, and logarithmic functions [13].

3.2. Nonzero test

Let I_x and I_y be the intervals over which x and y can respectively take their values. Let $f(I_x, I_y)$ be the interval obtained by replacing all operations that define the function $f(x, y)$ by their corresponding interval operations. If $0 \notin f(I_x, I_y)$, the curve $f(x, y) = 0$ is guaranteed not to pass through the region $I_x \times I_y$. Otherwise, we recursively subdivide the region and reject the cells through which the curve cannot pass.

However, if we simply replace the input variables by intervals and execute interval operations, the resulting intervals are usually too wide to restrict the function values with sufficient accuracy. A well-known method for improving the accuracy is the use of derivatives. Let I_m denote the midpoint of interval I. We define the *mean-value form* of function $f(x, y)$ as follows [13]:

$$\begin{aligned} M_f = {} & f(I_{x.m}, I_{y.m}) + f_x(I_x, I_y)(I_x - I_{x.m}) \\ & + f_y(I_x, I_y)(I_y - I_{y.m}) \end{aligned} \tag{3}$$

Here, real numbers are regarded as intervals of width 0, to which interval operations are applied. It can be shown that if $0 \notin M_f$, the function $f(x, y)$ is guaranteed not to take 0 in $I_x \times I_y$ [13, 16, 20].

3.3. Finding all solutions

Consider simultaneous nonlinear equations

$$f(x, y) = 0, \qquad g(x, y) = 0 \tag{4}$$

over $I_x \times I_y$. No solutions exist if evaluation of the intervals of $f(x, y)$ and $g(x, y)$ and their mean-value forms implies $f(x, y) \neq 0$ or $g(x, y) \neq 0$ in $I_x \times I_y$. Otherwise, define the *Krawczyk forms* K_x and K_y as follows:

$$\begin{aligned} \begin{pmatrix} K_x \\ K_y \end{pmatrix} = {} & \begin{pmatrix} I_{x.m} \\ I_{y.m} \end{pmatrix} - \boldsymbol{L} \begin{pmatrix} f(I_{x.m}, I_{y.m}) \\ g(I_{x.m}, I_{y.m}) \end{pmatrix} \\ & + (\boldsymbol{I} - \boldsymbol{L}\mathcal{D}) \begin{pmatrix} I_x - I_{x.m} \\ I_y - I_{y.m} \end{pmatrix}. \end{aligned} \tag{5}$$

Here, $\boldsymbol{I}$ denotes the 2 × 2 unit matrix, and $\mathcal{D}$ denotes the following *interval gradient matrix* with interval elements:

$$\mathcal{D} = \begin{pmatrix} f_x(I_x, I_y) & f_y(I_x, I_y) \\ g_x(I_x, I_y) & g_y(I_x, I_y) \end{pmatrix} \tag{6}$$

In Eq. (5), $\boldsymbol{L}$ is the inverse of the matrix consisting of the midpoints of the intervals that constitute the interval gradient matrix $\mathcal{D}$:

$$\boldsymbol{L} = \begin{pmatrix} f_x(I_x, I_y)_{.m} & f_y(I_x, I_y)_{.m} \\ g_x(I_x, I_y)_{.m} & g_y(I_x, I_y)_{.m} \end{pmatrix}^{-1} \tag{7}$$

If $K \cap I_x = \emptyset$ or $K_y \cap I_y = \emptyset$, Eqs. (4) are guaranteed to have no solutions in $I_x \times I_y$ [16, 20]. If $K_x \subset I_x$ and $K_y \subset I_y$, it can be shown that Eqs. (4) have a unique solution in $I_y \times I_y$ provided

$$\|\boldsymbol{I} - \boldsymbol{L}\mathcal{D}\|_\infty < 1 \tag{8}$$

Moreover, Newton iterations always converge to that solution [16, 20]. Here, the *Newton norm* $\|\mathcal{I}\|_\infty$ of an interval matrix $\mathcal{I}$ whose (ij) element is interval I_{ij} is defined by

$$\|\mathcal{I}\|_\infty = \max_i \sum_j \mathrm{maxabs}(I_{ij}) \tag{9}$$

where maxabs(I) denotes the endpoint of the interval I with a larger absolute value. If no conditions described above can apply, we subdivide the region and do the same test recursively.

4. Displaying Plane Curves

We display the curve $f(x, y) = 0$ as follows. First, we compute *all* critical points in the display region by solving Eqs. (1) by means of the interval analysis described in the preceding section. Then, we recursively subdivide the display region into four rectangular cells of half size until the cell size is below the resolution specified by the user. Then, those cells that contain critical points inside are further subdivided in such a way that the critical points are at the cell vertices. In each step of this subdivision, we evaluate the interval of the function $f(x, y)$ and its mean-value form and remove those cells through which the curve cannot pass. As a result, the subdivision is always restricted to the neighborhoods of the curve, saving the computation time considerably.

Once we have found cells through which the curve can pass, we exhaustively search for the points that satisfy $f(x, y) = 0$ on the cell boundaries. This reduces to solving an equation of a single variable, so we can find all the solutions by evaluating the interval, mean-value form, and Krawczyk form of the function on each side. This process is a one-dimensional version of the procedure described in the preceding section. The mean-value form and the Krawczyk form of a function $f(x)$ of a single variable are defined as follows:

$$M_f = f(I_{.m}) + f'(I)(I - I_{.m}) \tag{10}$$

$$K_f = I_{.m} - \frac{f(I_{.m})}{f'(I)_{.m}} + \left(1 - \frac{f'(I)}{f'(I)_{.m}}\right)(I - I_{.m}) \tag{11}$$

If $0 \notin f(I)$ or $0 \notin M_f$ or $K_f \cap I = \emptyset$, it is guaranteed that no solutions exist in I [16, 20]. If $K_f \subset I$, it can be shown that there exists a unique solution in I provided $|I - f'(I)/f'(I)_m| < 1$. Also, Newton iterations always converge to that solution [16, 20].

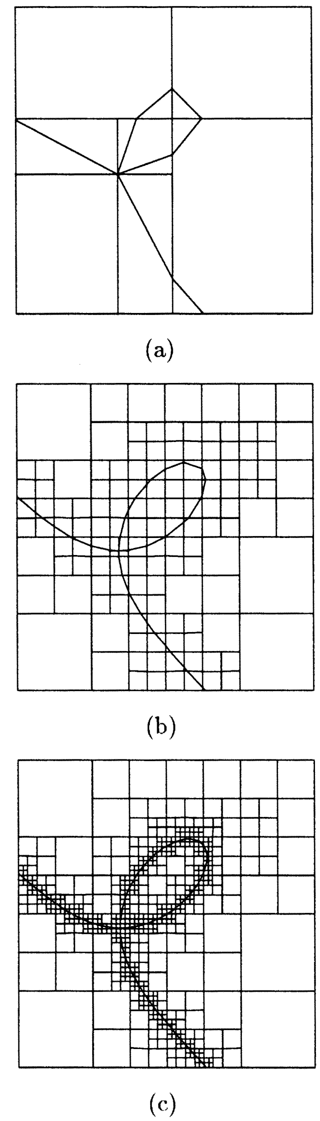


Fig. 1. Display of curve $x^3 + y^3 - 3xy = 0$ and the associated subdivision of the region. The resolution is (a) 1, (b) 1/8, (c) 1/64 the size of the display region.

In practice, however, it is more convenient to apply *regula falsi*: We compare the signs of the function at the endpoints of the interval in question and recursively subdivide it for exhausting the solutions. Our system adopts this strategy.

Next, we generate line segments that approximate the curve in each cell by connecting the points we have obtained on its cell boundary. Since it is guaranteed that no critical points exist inside the cell, a point that is neither an isolated point nor a singularity is connected to another point and no segments intersect inside the cell. If multiple candidates exist for this connection, we choose the one for which the function value $f(x, y)$ is the closest to zero along the candidate segments. Finally, we display the resulting segments.

To facilitate the execution of the above procedure, we classify the critical points into singularities and nonsingular critical points in advance. The singularities are further classified into isolated points and connected singularities. This is done by defining a small circle encircling the singularity in question, its radius being equal to the *limit resolution* (the finest resolution that users can specify). Restricting the domain of the function $f(x, y)$ to this circle and defining a function $f(\theta)$ of the angular variable θ, we solve $f(\theta) = 0$ by evaluating its interval, mean-value form, and Krawczyk form over $[0, 2\pi]$. We judge the point to be an isolated point if no solutions exist and a connected singularity otherwise.

Figure 1 shows the curve

$$x^3 + y^3 - 3xy = 0 \tag{12}$$

over $[-2.5, 3.2] \times [-2.5, 3.2]$ and the associated subdivision of the region. This curve has one crossing and one loop. In Fig. 1(a), the curve is drawn in the coarsest resolution: The size of the display region itself is used as the resolution. Since there exist a singularity and a critical point inside, the region is automatically subdivided in such a way that these points are at the vertices of the subdivided cells.

Figures 1(b) and 1(c) show the curve with the resolution being 1/8 and 1/64 the size of the display region, respectively. We can see that the cell subdivision is always restricted to the neighborhood of the curve. Figure 2 plots the computation time (in seconds) versus N, where the resolution is $1/N$ the size of the display region. It is seen that the computation time is approximately $O(N)$. Note that it would be $O(N^2)$ if the cells that the curve cannot pass through were not discarded by interval analysis.

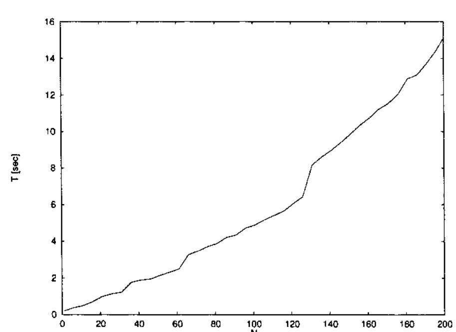


Fig. 2. Computation time (in seconds) of the curve in Fig. 1. The resolution is $1/N$ the size of the display region.

Figure 3 shows the curve

$$x^3 + x^2 + y^2 = 0 \tag{13}$$

over $[-2.5, 3.2] \times [-2.5, 3.2]$ and the associated subdivision of the region. This curve has one singularity (isolated point) and one critical point between the isolated point and the curve. See Ref. 10 for more complicated curves displayed similarly.

If we display the curve $f(x, y) - c = 0$ for a given function $f(x, y)$ and vary the value of the constant c, we obtain the contour curves of the surface $z = f(x, y)$. Since the critical points of $f(x, y) - c = 0$ do not depend on the value c, they need to be computed only once in advance. Figure 4 shows the contour curves of the surface

$$z = x^3 - 3x^2 + 3y^2 \tag{14}$$

over $[-2.5, 3.2] \times [-2.5, 3.2]$. We can see that the contour curves that pass through the singularity are displayed with correct topology.

5. Displaying Space Curves

Consider the space curve defined by the two equations

$$F(x, y, z) = 0, \quad G(x, y, z) = 0 \tag{15}$$

A *critical point* of this curve is a point such that

$$\begin{aligned} F_yG_z - F_zG_y &= 0, & F_zG_x - F_xG_z &= 0, \\ F_xG_y - F_yG_x &= 0, & FG &= 0 \end{aligned} \tag{16}$$

If the point satisfies $F = G = 0$ in addition, it is called a *singularity*, at which the surface defined by $F(x, y, z) = 0$ and the surface defined by $G(x, y, z) = 0$ are tangent to each other. Here, we are assuming that the two surfaces are smooth and their intersection does not pass through singularities, if any, of either surface.

As in the case of plane curves, we compute all the critical points inside the specified region before displaying the curve. This reduces to solving four equations of three variables, and the computation is similar to the case of two variables. First, we conduct the nonzero test to each equation separately, evaluating its interval, mean-value form, and Krawczyk forms. The Krawczyk forms for four equations of three variables are defined by straightforwardly extending the Krawczyk forms given in Eqs. (5) for two variables: The interval gradient matrix $\mathcal{D}$ given in Eq. (6) is extended to a 4×3 matrix; the matrix $\boldsymbol{L}$ in Eq. (7) is replaced

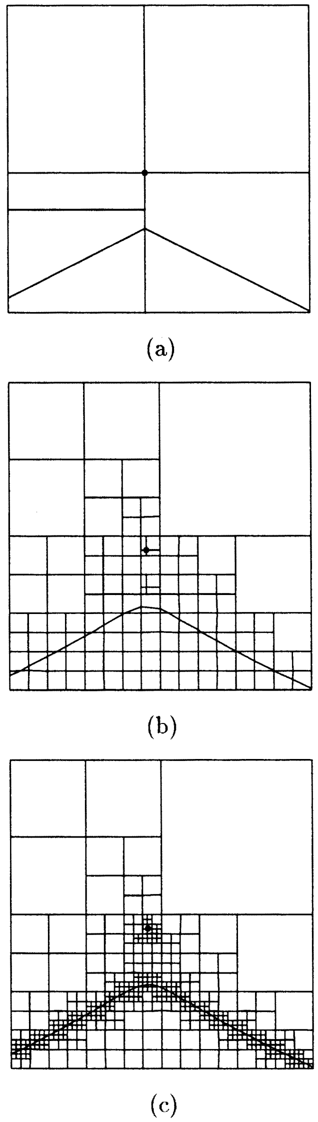


Fig. 3. Display of curve $x^3 + x^2 + y^2 = 0$ and the associated subdivision of the region. The resolution is (a) 1, (b) 1/8, (c) 1/64 the size of the display region.

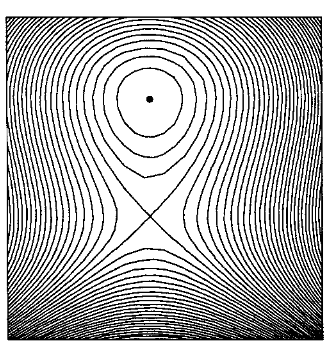


Fig. 4. Contours curves of the surface $z = x^3 - 3x^2 + 3y^2$.

by the (Moore–Penrose) generalized inverse of the matrix consisting of the midpoints of the interval elements of $\mathcal{D}$ [11]. Similarly, the Newton norm and the condition for the existence of the solution are straightforwardly generalized for four equations of three variables: The update equation for Newton iterations is now a set of four linear equations in three variables, so we compute the least-squares solution [11].

Having computed all the critical points, we recursively subdivide the region into eight rectangular cells of half size until the cell size is below the resolution specified by the user. Then, those cells that contain critical points inside are further subdivided in such a way that the critical points are at the cell vertices. In each step of this subdivision, we evaluate the intervals of the functions $F(x, y, z)$ and $G(x, y, z)$ and their mean-value forms and remove those cells through which the curve cannot pass, thereby restricting the subdivision to the neighborhoods of the curve.

For each of the cells through which the curve can pass, we search for the points that satisfy $F(x, y, z) = 0$ and $G(x, y, z) = 0$ on the cell surface. This reduces to solving two equations of two variables, so we can find all the solutions by evaluating the intervals, mean-value forms, and Krawczyk forms on each face.

Then, we generate line segments that approximate the curve in each cell by connecting the computed points on its surface in such a way that a nonsingular point is connected to another point and no segments intersect inside the cell. If multiple candidates exist, we choose the one for which the function values $F(x, y, z)$ and $G(x, y, z)$ are the closest to zero along the candidate segments. Finally, we project the resulting segments onto the two-dimensional plane (*retina*) associated with a given viewpoint and display this projection.

As in two dimensions, we classify the critical points into singularities and nonsingular critical points in advance. The singularities are further classified into isolated points and connected singularities. This is done by defining a sphere enclosing the singularity in question, its radius being equal to the limit resolution. Restricting the domains of the functions $F(x, y, z)$ and $G(x, y, z)$ onto this sphere and defining functions $F(\theta, \phi)$ and $G(\theta, \phi)$ of the spherical variable (θ, ϕ), we solve $F(\theta, \phi) = 0$ and $G(\theta, \phi) = 0$ by evaluating their intervals, mean-value forms, and Krawczyk forms over $[0, \pi] \times [0, 2\pi]$. We judge the point to be an isolated point if no solutions exist and a connected singularity otherwise.

As in the case of plane curves, no isolated points or crossings exist inside the cells since singularities are exhaustively computed and located at the cell vertices by construction. The only difference is that, in theory, small loops could be undetected: They are usually detected because two critical points exist in the neighborhood of a small loop, but they could be overlooked, depending on the positions of the critical points and the cell size. This could be overcome by incorporating ad hoc treatments, but to avoid complication, we do not go into the details. At any rate, this problem is resolved by choosing a reasonably small cell size.

The two ellipsoids defined by

$$3x^2 + y^2 + 2z^2 = 1, \quad 3x^2 + 2y^2 + z^2 = \beta \qquad (17)$$

intersect along two closed curves when $\beta \neq 1$. When $\beta = 1$, there arise two singularities, at which the two intersection curves intersect. Figures 5(a), 5(b), and 5(c) show the singular intersection in the region $[-1.2, 1.3] \times [-1.2, 1.3] \times [-1.2, 1.3]$, the resolution being 1/2, 1/8, and 1/32 the size of the region, respectively. In Fig. 5(c), the occluding contours of the two ellipsoids are also displayed to help visual interpretation (we will describe the procedure later). We can see that the singularities are correctly displayed irrespective of the resolution.

Figures 6(a), 6(b), and 6(c) show the corresponding curves for $\beta = 1 + 10^{-6}$. Although the intersection curves may appear to be crossing, their magnification would reveal that they do not because the line segments defining them do not intersect. We can also see that the topology is correct irrespective of the resolution. Figure 7 plots the computation time (in seconds) versus N for computing Fig. 5, where the resolution is $1/N$ the size of the region. It is seen that the computation time is approximately $O(N)$; it would be $O(N^2)$ if the cells that the curve cannot pass through were not discarded by interval analysis.

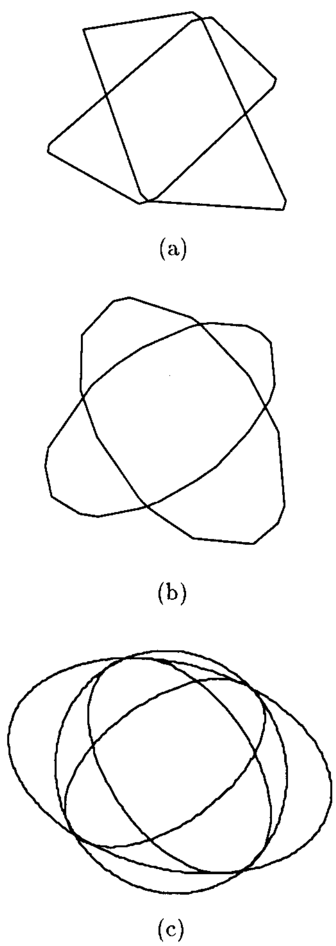


Fig. 5. Intersection of two ellipsoids (with a singularity). The resolution is (a) 1/2, (b) 1/8, (c) 1/32 the size of the display region.

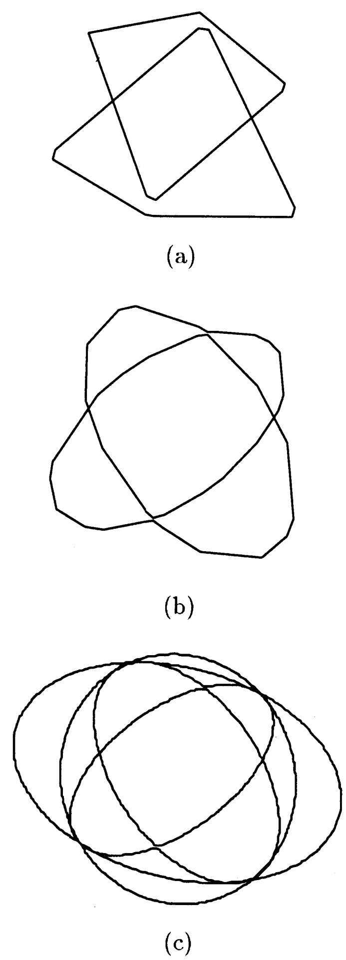


Fig 6. Intersection of two ellipsoids (without a singularity). The resolution is (a) 1/2, (b) 1/8, (c) 1/32 the size of the display region.

6. Displaying Surfaces

As pointed out in the Introduction, a surface can be most naturally displayed by its occluding contours and rims.

6.1. Graph representation

Consider the surface

$$z = f(x, y) \tag{18}$$

Its rims can be easily generated by dividing the circumference of the display region into intervals of width being equal to the resolution, computing the height z at each dividing point, and defining the line segments by connecting the resulting points in three dimensions. The occluding contours of this surface are a projection of the contour generator in three dimensions onto the two-dimensional retina. Let V: (v_1, v_2, v_3) be the viewpoint. A point P: (x, y, z) is on the contour generator if and only if the tangent plane to this surface at P passes through V. This condition is written as follows [9]:

$$(x - v_1)f_x + (y - v_2)f_y - (f - v_3) = 0 \tag{19}$$

We first generate the line segments that approximate the plane curve defined by this equation in the xy plane by the method described in Section 4. We then define the corresponding line segments in three dimensions by computing the height z at their endpoints. Finally, we project the line segments onto the retina associated with the viewpoint V.

Figures 8(a) and 8(b) compare the wireframe representation (the intersections of the surface with equidistant planes having constant x and y coordinates) and the occluding contour representation of the surface

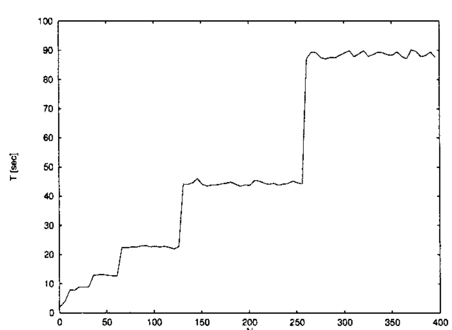


Fig. 7. Computation time (in seconds) of the curve in Fig. 5. The resolution is $1/N$ the size of the display region.

$$z = x^2 - x^2y + y^3 \tag{20}$$

The resolution is 1/8 the size of the display region. In Fig. 8(b), we can observe singularities of the occluding contours called *lips* [8, 9, 12]; they are displayed with correct topology. This type of singularity is very difficult to visualize by the standard computer graphics techniques based on wireframes and their surface rendering using shading and texture.

6.2. Implicit representation

Consider the surface

$$F(x, y, z) = 0 \tag{21}$$

Let V: (v_1, v_2, v_3) be the viewpoint. A point P: (x, y, z) is on the contour generator if and only if the tangent plane to this surface at P passes through V. This condition is written as follows [9]:

$$F = 0, \quad (x - v_1)F_x + (y - v_2)F_y + (z - v_3)F_z = 0 \tag{22}$$

The contour generator is the space curve defined by these two equations. The occluding contour can be displayed by generating the line segments that approximate the contour generator by the method described in Section 5 and projecting them onto the retina associated with the viewpoint V.

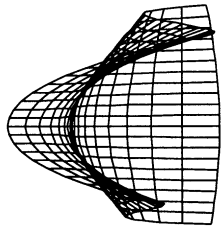
Figures 9(a) and 9(b) compare the wireframe representation (the intersections of the surface with equidistant planes having constant x, y, and z coordinates) and the occluding contour representation of the surface

$$\begin{aligned} &23x^4 + x^2y^2 - 37x^2y - 2xy^2 - 15x^2 \\ &\quad -2xy + 16y^2 + 16z^2 - x + 16y = 0 \end{aligned} \tag{23}$$

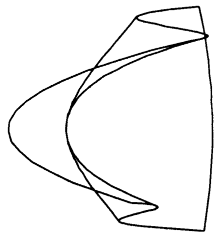
In Fig. 9(b), we can observe singularities of the occluding contours called *cusps* [8, 9, 12]; they are displayed with correct topology.

6.3. Hidden line removal

Figures 8 and 9 depict transparent surfaces, but it is easy to make them look opaque: All we need to do is remove invisible contours hidden by the surface. A point P on surface $F(x, y, z) = 0$ is visible if and only if the equation $F = 0$ has a solution on the line segment that connects P and the viewpoint V. Thus, testing the visibility reduces to solving an equation of a single variable. The solutions can be obtained by evaluating the interval, mean-value form, and Krawczyk form of F. For each line segment that constitutes the contour generator, we test the visibility of its

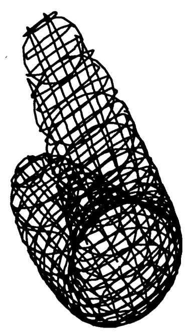


(a)

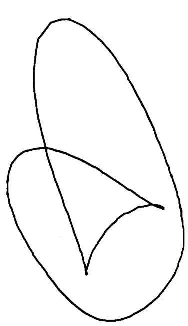


(b)

Fig. 8. Display of surface $z = x^2 - x^2y + y^3$.
(a) Wireframes. (b) Rims and occluding contours.

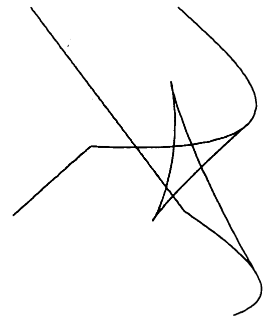


(a)

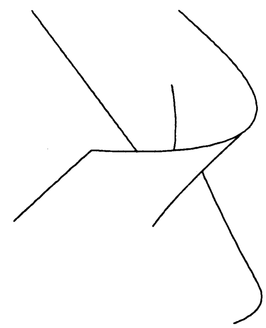


(b)

Fig. 9. Display of surface $23x^4 + x^2y^2 - 37x^2y - 2xy^2 - 15x^2 - 2xy + 16y^2 + 16z - x + 16y = 0$.
(a) Wireframes. (b) Occluding contours.



(a)



(b)

Fig. 10. Occluding contours of surface $z = xy + y^4$.
(a) All contours. (b) Hidden line removal.

endpoints and remove it if the two endpoints are both invisible. If only one endpoint is visible, we divide the segment and apply the same test recursively.

Figures 10(a) and 10(b) compare the occluding contour representations of the surface

$$z = xy + y^4 \tag{24}$$

with and without hidden occluding contours. As we can see, humans can easily and correctly understand the 3D shape of the surface by merely seeing its occluding contours, and hidden line removal greatly facilitates this understanding.

7. Concluding Remarks

We have presented an efficient method for displaying curves and surfaces defined by implicit equations in such a way that the topology (connection) is correct in whatever resolution. We pointed out the problems arising from the traditional curve tracing schemes and proposed an efficient method that combines recursive subdivision of the display region with interval analysis techniques. The topology of the curve is preserved because we compute all critical points that determine the topology of the curve by evaluating the intervals, mean-value forms, and Krawczyk forms. We have experimentally confirmed that our method indeed preserves the topology in all resolutions.

Unlike the method of Taniguchi and Sugihara [19], our method does not require complicated procedures for symbolic algebra, such as computation of resolvants, for detecting crossings, branchings, and loops of algebraic curves. In fact, our method does not even require the curves to be algebraic. We applied our method to displaying surfaces by drawing rims and occluding contours. This representation is more natural for human visual perception than wireframes with surface rendering.

Our system defines interval objects and their methods in C++ language. As a result, operations of intervals are performed as if they are for real numbers once the variables and constants are declared to be of type "interval." Our system automates differentiation of functions by using symbolic manipulations. The resulting procedures are converted into interval programs. Our system was implemented on a PC/AT convertible machine with Pentium Pro 200MHz as the CPU, Linux 2.0 (Slackware 3.5) as the OS, and GCC 2.7.2.3 as the compiler.

Acknowledgments. The authors thank Kiyotaka Yamamura of Chuo University for providing us with materials about interval analysis, and Yasushi Kanazawa of Toyohashi University, Koichi Kubota of Chuo University, and Toshinobu Yoshida of the University of Electro-Communication for providing us with materials about automatic differentiation. This work was supported in part by the Ministry of Education, Science, Sports and Culture, Japan under a Grant in Aid for Scientific Research C(2) (No. 11680377).

REFERENCES

1. Arnol'd VI. Catastrophe Theory (third revised and expanded edition). Springer; 1992.
2. Bruce JW. Seeing—The mathematical viewpoint. Math Intell 1984;6:18–25.
3. Cipolla R. Active visual reference of surface shape. Springer; 1995.
4. Farin G. Curves and surfaces or computer-aided geometric design: A practical guide (3rd edition). Academic Press; 1993.
5. Gray A. Modern differential geometry of curves and surfaces. CRC Press; 1993.
6. Hosaka M. Modeling of curves and surfaces in CAD/CAM. Springer-Verlag; 1992.
7. Kanazawa Y, Kashiwagi M, Oishi S. An algorithm of finding all solutions with guaranteed accuracy for parameter dependent nonlinear equations. Trans IEICE 1997;J80-A:920–925. (in Japanese)
8. Kanatani K. What surfaces look like: Searching for 'butterflies', 'geese' and 'gulls.' bit 1997;29:102–103. (in Japanese)
9. Kanatani K. Mathematics of curves and surfaces for CAD. Kyoritsu Shuppan; 1998. (in Japanese)
10. Kanazawa Y, Kanatani K. Displaying planar curves with correct topology in all resolutions. IPSJ SIG Notes, 99-CG-94, p 37–42, 1999. (in Japanese)
11. Kanazawa Y, Kanatani K. Displaying intuitively correct space curves and surfaces. Tech Rep IEICE, AIE98/KBSE98, p 37–42, 1999.
12. Koenderink JJ. Solid shape. MIT Press; 1990.
13. Moore RE. Methods and applications of interval analysis. SIAM; 1979.
14. Morimoto K, Yamaguchi K. Recursive subdivision methods and subspace classification: The extension to approximation of curves in small subspace. Trans IEICE 1997;J80-D-II:569–578. (in Japanese)
15. Nishio K, Kobori K, Kutsuwa T. A conversion of spatial partitioning model into boundary representations using extended octree. Trans Inf Process Soc Japan 1997;38:1554–1565.
16. Oishi S. Analytical approaches to nonlinear phenomena [1]: Numerical analysis of nonlinear phenomena with guaranteed accuracy (1). IEICE Mag 1996;79:162–168.
17. Sasaki T, Motoyoshi F, Watanabe S. Computer algebra systems. Shokodo; 1986. (in Japanese)
18. Sugihara K. Mathematics of computer graphics. Kyoritsu Shuppan; 1995. (in Japanese)
19. Taniguchi Y, Sugihara K. A robust method for tracing algebraic curves. Trans Inf Process Soc Japan 1992;33:1245–1253. (in Japanese)
20. Yamamura K. Analytical approaches to nonlinear phenomena [V]: Numerical solution of nonlinear equations. J IEICE 1996;79:740–745. (in Japanese)
21. Yamamura K, Tokue A, Kawata H. Finding all solutions of nonlinear resistive circuits by interval analysis. Trans IEICE 1996;J79-A:1692–1699. (in Japanese)

AUTHORS (from left to right)

Yoshiyuki Kanazawa received his B.S. and M.S. degrees in computer science in 1997 and 1999, respectively, from Gunma University, where he studied computer vision and computer graphics. In 1999 he joined Sharp, Ltd. Currently, he is engaged in research and development of liquid crystal display projector systems.

Kenichi Kanatani (member) received his Ph.D. degree in applied mathematics from the University of Tokyo in 1979. He is currently a professor of computer science at Gunma University. He is the author of *Group-Theoretical Methods in Image Understanding* (Springer, 1990), *Geometric Computation for Machine Vision* (Oxford, 1993), and *Statistical Optimization for Geometric Computation* (Elsevier, 1996). He is a member of IEICE. For details, see http://www.ail.cs.gunma-u.ac.jp/~kanatani/e