

Bisector Curves of Planar Rational Curves*

Gershon Elber

Department of Computer Science
Technion, Israel Institute of Technology
Haifa 32000, Israel
E-mail: gershon@cs.technion.ac.il

and

Myung-Soo Kim

Department of Computer Science
POSTECH
Pohang 790-784, South Korea
E-mail: mskim@postech.ac.kr

Abstract

This paper presents a simple and robust method for computing the bisector of two planar rational curves. We represent the correspondence between the foot points on two planar rational curves $C_1(t)$ and $C_2(r)$ as an implicit curve $\mathcal{F}(t, r) = 0$, where $\mathcal{F}(t, r)$ is a bivariate polynomial B-spline function. Given two rational curves of degree m in the xy -plane, the curve $\mathcal{F}(t, r) = 0$ has degree $4m - 2$, which is considerably lower than that of the corresponding bisector curve in the xy -plane.

Key Words: Bisector, skeleton, Voronoi diagram, pocket machining, zero-set, subdivision.

*This research was supported in part by the Korean Ministry of Science and Technology under Grant 97-NS-01-05-A-02-A of STEP 2000, by KOSEF (Korea Science and Engineering Foundation) under Grant 96-0100-01-01-2, and by the Fund for Promotion of Research at the Technion, Haifa, Israel.

1 Introduction

Given a planar region bounded by freeform curve segments, the set of all interior points that have equal minimum distance from (at least) two different boundary points is called the *skeleton* of the planar region. When we decompose the boundary curves at each extreme curvature point, the skeleton is composed of segments of *bisector curves*, for some pairs of boundary curves C_i and C_j , ($i \neq j$). Similarly, given a number of disjoint planar regions, their *Voronoi diagram* is defined as the set of points which are equidistant from (at least) two different regions. The Voronoi diagram also consists of bisector curves, for some pairs of boundary curves of the planar regions.

The skeleton and Voronoi diagram have important applications in NC pocket machining, finite element mesh generation, and collision-avoidance motion planning. In particular, Persson [15] shows that the skeleton structure can be employed quite effectively in NC pocket machining. Held [10] presents an algorithm that constructs the skeleton of a planar region, while discussing important issues in implementation and also in NC machining applications. Both Persson [15] and Held [10] consider planar regions bounded by line segments and circular arcs, in which case the bisector curves are limited to lines and conics.

To construct a skeleton or a Voronoi diagram, we need to determine an arrangement of bisector curves in the plane. This task is quite non-trivial using floating-point arithmetic, especially when the bisector curves include conics and even higher degree curves. Lavender et al. [14] suggest a subdivision technique (based on interval arithmetic) to construct the global structure of a Voronoi diagram. This method is general in the sense that it can handle arbitrary set-theoretic objects in any dimensions. Hoffmann [12] applies an image-processing technique to the construction of a skeleton, which is simple to implement. Both the input region and the output skeleton are represented as digital images. For planar regions with no holes, Chou [2] suggests an algorithm that is based on a tree structure of the skeleton. The construction starts at leaf nodes and proceeds in a bottom-up fashion.

Given a set of planar regions bounded by rational curves, we consider the construction of the skeletons and the Voronoi diagram of these planar regions. Once the global structure of a skeleton or a Voronoi diagram is constructed, one needs to accurately approximate the bisector curves since they are non-rational curves, in general [8]. In this paper, we focus on the issues related to the representation and construction of the bisectors of rational curves, while assuming the global structure of a skeleton or a Voronoi diagram is provided by one of the methods [2, 12, 14] discussed above. The constructor of the bisector curve is a basic building block that is crucial for the construction of a skeleton or a Voronoi diagram. Consequently, the construction of bisectors (of rational curves) has attracted considerable research attention in the past. We briefly review some representative results.

Hoffmann and Vermeer [13] formulate the bisector curve in terms of a system of polyno-

mial equations using some auxiliary variables. By eliminating these auxiliary variables, an implicit equation of the bisector curve can be obtained. However, the process of variable elimination is slow and generates an algebraic bisector curve of high degree. Hoffmann [11] also suggests using a dimensionality paradigm in which each polynomial equation is geometrically represented as a hyper-surface. The bisector curve of two planar rational curves can be computed via the intersection of three hyper-surfaces in \mathbb{R}^4 . The multiple intersection in \mathbb{R}^4 is inefficient, compared with that of intersecting two surfaces in \mathbb{R}^3 .

Choi [1] transforms the problem of computing a bisector curve into that of intersecting two developable surfaces in \mathbb{R}^3 . He also shows that the intersection curve of two developable surfaces can be computed in an efficient and robust way. Nevertheless, it is non-trivial to implement the algorithm of Choi [1]. Recently, Heo et al. [5] suggest a simple algorithm that can intersect two ruled surfaces efficiently and robustly, while dealing with all possible degenerate cases. Since developable surface is a special type of ruled surface, we may apply this result to the bisector problem.

Farouki and Johnstone [7] show that the bisector of a point and a rational curve in the same plane is a rational curve. Given two planar rational curves, Farouki and Johnstone [8] interpret the bisector as the envelope curve of a one-parameter family of rational point/curve bisectors. The curve/curve bisector is non-rational, in general. Therefore, Farouki and Johnstone [8] approximate the bisector curve with a sequence of discrete points. Farouki and Ramamurthy [9] develop a more precise algorithm that approximates the bisector curve with a sequence of curve segments. The approximation error can be reduced within an arbitrary bound by adaptive refinement.

To generate each point on a curve/curve bisector, the two algorithms [8, 9] construct a point/curve bisector and then intersect it with a line. Untrimmed point/curve bisectors are easy to construct [7]. The intersection points are computed numerically by solving polynomial equations of degree $3n - 1$ or $4n - 2$ for polynomial or rational input curves of degree n [8]. Approximation error is also estimated numerically [9]. The numerical nature of the procedures makes the algorithms computationally intensive. Nevertheless, the algorithm of Farouki and Ramamurthy [9] is robust in the sense that it will never miss small loops provided that all numerical computations are carried out with sufficient precision (see also the Ph.D. Thesis of Ramamurthy [16] for more details and other related results). In fact, the algorithm [9] provides a robust bisector construction for arbitrary planar curves, not limited to rational curves. On the other hand, no symbolic processing is applied to input curves and the entire bisector curve has no closed-form representation.

In this paper, we suggest a new (symbolic) representation scheme for planar bisector curves, which allows an efficient and robust implementation of a bisector curve construction based on B-spline subdivision techniques. A priority is given to the development of a

method which is simple while robustness is drawn from the B-spline representation's subdivision and convex hull containment properties. All the algorithms and examples presented in this paper were implemented and created with the aid of tools available in the IRIT [6] solid modeling system, developed at the Technion, Israel.

The rest of this paper is organized as follows. In Section 2, we outline the basic approach of this paper. Section 3 presents three different methods to compute the bivariate functions, $\mathcal{F}_i(t, r)$, ($i = 1, 2, 3$). The zero-set of each function $\mathcal{F}_i(t, r)$ defines the bisector curve of two rational curves in the plane. Some experimental results are demonstrated in Section 4. Finally, in Section 5, we conclude the paper.

2 Basic Idea

Given two planar curves $C_1(t)$ and $C_2(r)$ represented in polynomial/rational forms, we symbolically compute a bivariate polynomial function $\mathcal{F}(t, r)$, the zero-set of which: $\mathcal{F}(t, r) = 0$ corresponds to the bisector curve. The formulation of $\mathcal{F}(t, r)$ is based on a symbolic substitution of polynomial/rational functions of t and r into a simple polynomial expression, in the tr -plane. In contrast to this, the conventional approach [13] generates the bisector as an implicit algebraic curve in the xy -plane by eliminating some auxiliary variables, which takes more computation time. Moreover, the implicit curve equation thus generated has a much higher degree.

The zero-set of $\mathcal{F}(t, r) = 0$ is an implicit algebraic curve in the tr -plane. In general, tracing along an implicit curve is a non-trivial task. That is, it is difficult to detect how many disconnected components an implicit curve has and then to generate a starting point for tracing along each connected component. Consequently, it is difficult to develop a robust algorithm based on a direct curve tracing. In this paper, we take a subdivision-based approach which guarantees the robustness of our algorithm.

Given a bivariate B-spline polynomial function $\mathcal{F}(t, r)$, we consider its graph surface $\mathcal{S}(t, r) = (t, r, \mathcal{F}(t, r))$ as a polynomial B-spline surface. The extraction of the zero-set of $\mathcal{F}(t, r) = 0$ can be computed quite efficiently and robustly using the subdivision and the convex hull containment properties of the B-spline representation. That is, we subdivide a B-spline surface S into four subpatches (i.e., the (t, r) -domain of S is subdivided into four subdomains). Then, the control points of each subpatch are tested whether they are above or below the tr -plane. We purge away all subpatches that are found to be completely above or completely below the tr -plane and recursively subdivide the remaining subpatches.

The main computational overhead of this method stems from the subdivision of the B-spline function $\mathcal{F}(t, r)$. Therefore, it is quite important to generate the lowest degree B-spline function $\mathcal{F}(t, r)$ the zero-set of which contains the bisector curve. In this paper, we suggest three different methods to generate such a B-spline function. A zero-set may

contain some redundant bisector curve segments. However, we will pay no attention to the elimination of these redundancies since this elimination is much easier when we use the global structures available from some other methods [12, 14].

3 Bivariate Functions

Let $C_1(t) = (x_1(t), y_1(t))$ and $C_2(r) = (x_2(r), y_2(r))$ be two planar regular rational curves with C^1 -continuity. In this section, we consider different ways of deriving the bivariate functions $\mathcal{F}_i(t, r)$ so that their zero-sets contain the bisector curve of $C_1(t)$ and $C_2(r)$. We also describe how to construct the bisector curve itself when such a zero-set is available (see Section 3.5).

3.1 Function $\mathcal{F}_1(t, r)$

Let $N_1(t) = (-y_1'(t), x_1'(t))$ and $N_2(r) = (-y_2'(r), x_2'(r))$ denote the *unnormalized* normal fields of $C_1(t)$ and $C_2(r)$, respectively. The intersection point of the normal lines of $C_1(t)$ and $C_2(r)$ is given as follows:

$$C_1(t) + N_1(t)\alpha = C_2(r) + N_2(r)\beta,$$

for some α and β . We then have the following two equations in two unknowns, α and β :

$$\begin{aligned} x_1(t) - y_1'(t)\alpha &= x_2(r) - y_2'(r)\beta, \\ y_1(t) + x_1'(t)\alpha &= y_2(r) + x_2'(r)\beta. \end{aligned}$$

Using Cramer's rule, we obtain the solutions of α and β as bivariate rational functions:

$$\begin{aligned} \alpha &= \alpha(t, r) = \frac{\begin{vmatrix} x_2(r) - x_1(t) & y_2'(r) \\ y_2(r) - y_1(t) & -x_2'(r) \end{vmatrix}}{\begin{vmatrix} -y_1'(t) & y_2'(r) \\ x_1'(t) & -x_2'(r) \end{vmatrix}}, \\ \beta &= \beta(t, r) = \frac{\begin{vmatrix} -y_1'(t) & x_2(r) - x_1(t) \\ x_1'(t) & y_2(r) - y_1(t) \end{vmatrix}}{\begin{vmatrix} -y_1'(t) & y_2'(r) \\ x_1'(t) & -x_2'(r) \end{vmatrix}}. \end{aligned} \tag{1}$$

The two bivariate functions $\alpha(t, r)$ and $\beta(t, r)$ provide a bivariate parameterization of the intersection point:

$$\begin{aligned} P(t, r) &= C_1(t) + N_1(t)\alpha(t, r) \\ &= C_2(r) + N_2(r)\beta(t, r). \end{aligned}$$

For $P(t, r)$ to be on the bisector curve of $C_1(t)$ and $C_2(r)$, it must be at an equal distance from $C_1(t)$ and $C_2(r)$:

$$\begin{aligned}
& \|P(t, r) - C_1(t)\| \\
&= \|(C_1(t) + N_1(t)\alpha(t, r)) - C_1(t)\| \\
&= \|(C_2(r) + N_2(r)\beta(t, r)) - C_2(r)\| \\
&= \|P(t, r) - C_2(r)\|,
\end{aligned}$$

which immediately reduces to

$$\|N_1(t)\alpha(t, r)\| = \|N_2(r)\beta(t, r)\|.$$

By squaring this equation and substituting $\alpha(t, r)$ and $\beta(t, r)$ from Equation (1) into the resulting equation, we get

$$\begin{aligned}
0 &= \langle N_1(t), N_1(t) \rangle \alpha(t, r)^2 - \langle N_2(r), N_2(r) \rangle \beta(t, r)^2 \\
&= \langle N_1(t), N_1(t) \rangle \frac{\begin{vmatrix} x_2(r) - x_1(t) & y_2'(r) \\ y_2(r) - y_1(t) & -x_2'(r) \end{vmatrix}^2}{\begin{vmatrix} -y_1'(t) & y_2'(r) \\ x_1'(t) & -x_2'(r) \end{vmatrix}^2} \\
&\quad - \langle N_2(r), N_2(r) \rangle \frac{\begin{vmatrix} -y_1'(t) & x_2(r) - x_1(t) \\ x_1'(t) & y_2(r) - y_1(t) \end{vmatrix}^2}{\begin{vmatrix} -y_1'(t) & y_2'(r) \\ x_1'(t) & -x_2'(r) \end{vmatrix}^2}. \tag{2}
\end{aligned}$$

Even if the two curves are regular, the denominator of Equation (2) would vanish when the tangents/normals of the two curves become parallel or opposite at some parameter values of t and r . Geometrically speaking, this condition implies that the two normal lines are parallel. Two parallel lines in the plane may have no intersection or otherwise they completely overlap. Assume that $N_1(t)$ and $N_2(r)$ are neither parallel nor opposite. Then, the denominator of Equation (2) cannot vanish and the zero-set constraint resulting from Equation (2) reduces to:

$$\begin{aligned}
0 &= \mathcal{F}_1(t, r) \\
&= \langle N_1(t), N_1(t) \rangle \begin{vmatrix} x_2(r) - x_1(t) & y_2'(r) \\ y_2(r) - y_1(t) & -x_2'(r) \end{vmatrix}^2 \\
&\quad - \langle N_2(r), N_2(r) \rangle \begin{vmatrix} -y_1'(t) & x_2(r) - x_1(t) \\ x_1'(t) & y_2(r) - y_1(t) \end{vmatrix}^2 \\
&= \left(x_1'^2(t) + y_1'^2(t)\right) \left((x_1(t) - x_2(r))x_2'(r) + (y_1(t) - y_2(r))y_2'(r)\right)^2 \\
&\quad - \left(x_2'^2(r) + y_2'^2(r)\right) \left((x_1(t) - x_2(r))x_1'(t) + (y_1(t) - y_2(r))y_1'(t)\right)^2. \tag{3}
\end{aligned}$$

The solution of Equation (3) represents the set of all pairs of curve points on $C_1(t)$ and $C_2(r)$ that share a bisector point. The corresponding points on $C_1(t)$ and $C_2(r)$ are called the *foot points* of the bisector point.

Note that Equation (3) is equivalent to Equation (2) under the condition that the tangents/normals of the two curves are neither parallel nor opposite. When the two tangents/normals are parallel or opposite, Equation (2) is undefined. Nevertheless, even in this case, Equation (3) is well defined and satisfies the condition of $\mathcal{F}_1(t, r) = 0$. In fact, Equation (3) has the following component:

$$x'_1(t)y'_2(r) - x'_2(r)y'_1(t) = 0, \quad (4)$$

which implies that the tangents/normals of the curves are parallel or opposite. Note that the solutions of Equation (4) generate bisector points at infinity when the two normal lines of the curves are parallel, but do not overlap each other. Moreover, when the normal lines overlap in the same line, the pair (t, r) for a bisector point (in the real, affine plane) must satisfy the following two additional constraints:

$$\begin{aligned} x'_1(t)(x_1(t) - x_2(r)) + y'_1(t)(y_1(t) - y_2(r)) &= 0, \\ x'_2(r)(x_1(t) - x_2(r)) + y'_2(r)(y_1(t) - y_2(r)) &= 0. \end{aligned} \quad (5)$$

Among the solutions of Equation (4), only the common solutions with Equation (5) contribute to the bisector curve.

3.2 Function $\mathcal{F}_2(t, r)$

Equation (3) can also be derived based on a different geometric interpretation. The bisector point, $P(t, r)$, of two curves $C_1(t)$ and $C_2(r)$ satisfies the following angular relationship (see Figure 1):

$$\langle C_1(t) - C_2(r), \vec{T}_1(t) \rangle = \langle C_2(t) - C_1(r), \gamma \vec{T}_2(r) \rangle,$$

where γ is either -1 or $+1$, and \vec{T}_i denotes a *unit* tangent vector. By squaring both sides of this equation, we get

$$\begin{aligned} 0 &= \left(\langle C_1(t) - C_2(r), \vec{T}_1(t) \rangle \right)^2 - \left(\langle C_2(t) - C_1(r), \gamma \vec{T}_2(r) \rangle \right)^2 \\ &= \left(\frac{\langle C_1(t) - C_2(r), T_1(t) \rangle}{\|T_1(t)\|} \right)^2 - \left(\frac{\langle C_1(t) - C_2(r), T_2(r) \rangle}{\|T_2(r)\|} \right)^2 \\ &= \frac{\left((x_1(t) - x_2(r))x'_1(t) + (y_1(t) - y_2(r))y'_1(t) \right)^2}{x_1'^2(t) + y_1'^2(t)} \\ &\quad - \frac{\left((x_1(t) - x_2(r))x'_2(r) + (y_1(t) - y_2(r))y'_2(r) \right)^2}{x_2'^2(r) + y_2'^2(r)}, \end{aligned} \quad (6)$$

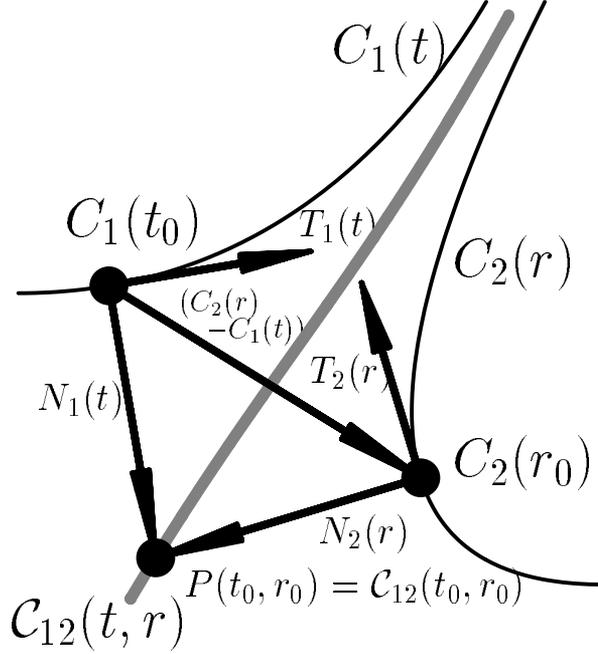


Figure 1: The bisector, $\mathcal{C}_{12}(t, r)$, of two curves, $C_1(t)$ and $C_2(r)$, constructed from the equiangular relationship between the vector $\pm(C_1(t) - C_2(r))$ and the two unit normals, $\vec{N}_1(t)$ and $\vec{N}_2(r)$, or the vector $\pm(C_1(t) - C_2(r))$ and the two unit tangents, $\vec{T}_1(t)$ and $\vec{T}_2(r)$.

where $T_1(t) = (x'_1(t), y'_1(t))$ and $T_2(r) = (x'_2(r), y'_2(r))$ denote the *unnormalized* tangent fields of $C_1(t)$ and $C_2(r)$, respectively. Since the two curves $C_1(t)$ and $C_2(r)$ are regular, we have

$$(x_1'^2(t) + y_1'^2(t))(x_2'^2(r) + y_2'^2(r)) \neq 0. \quad (7)$$

Thus, Equation (6) is equivalent to Equation (3).

Equation (6) actually *matches* the points on $C_1(t)$ and $C_2(r)$ in such a way that the two angles between the line through both $C_1(t)$ and $C_2(r)$ and the two tangents of the curves are equal. This set of matching points forms a superset of the real bisector curve since there are two different tangent directions for each curve. Therefore, Equation (6) has a redundant component which is not on the real bisector curve. Note that all (t, r) -pairs corresponding to parallel/opposite curve tangents also satisfy Equation (6) even if the tangents of the curves are not orthogonal to the difference vector $C_1(t) - C_2(r)$. Hence, the redundant component of Equation (6) is exactly the same as the curve component given in Equation (4). In fact, this is also an immediate consequence from the equivalence of Equation (3) and Equation (6).

An alternative solution to the bisector of two planar curves can be formulated by equalizing the angles between the *normals* of the two curves instead of their tangents:

$$0 = \hat{\mathcal{F}}_2(t, r)$$

$$= \langle C_1(t) - C_2(r), \vec{N}_1(t) \rangle^2 - \langle C_1(t) - C_2(r), \vec{N}_2(r) \rangle^2.$$

When we multiply this equation with the term in Equation (7), we get

$$\begin{aligned} 0 &= \mathcal{F}_2(t, r) \\ &= (x_1'^2(t) + y_1'^2(t))(x_2'^2(r) + y_2'^2(r))\hat{\mathcal{F}}_2(t, r). \end{aligned}$$

While $\hat{\mathcal{F}}_2$ is a rational function, in general, it should be pointed out that \mathcal{F}_2 is a polynomial function when both $C_1(t)$ and $C_2(r)$ are polynomial curves.

3.3 Function $\mathcal{F}_3(t, r)$

The bisector curve of two rational curves in \mathbb{R}^2 is non-rational, in general. Quite surprisingly, Elber and Kim [4] showed that the bisector surface of two rational space curves in \mathbb{R}^3 is indeed a rational surface, except for the degenerate case in which the two space curves are coplanar. They also presented a constructive algorithm for the rational parameterization of the bisector surface. In this subsection, we show that the constraints for constructing the rational bisector surface in \mathbb{R}^3 can be reduced to the formulation of a bivariate rational function, the zero-set of which defines the bisector of two planar curves.

Given two space curves $C_1(t)$ and $C_2(r)$ in \mathbb{R}^3 , Elber and Kim [4] formulated the following three (linear in Q) conditions that a bisector point Q must satisfy:

$$\langle Q - C_1(t), C_1'(t) \rangle = 0, \quad (8)$$

$$\langle Q - C_2(r), C_2'(r) \rangle = 0, \quad (9)$$

$$\left\langle Q - \frac{C_1(t) + C_2(r)}{2}, C_1(t) - C_2(r) \right\rangle = 0. \quad (10)$$

The first two conditions of Equations (8)–(9) imply that the vector from each curve to the bisector point is orthogonal to the tangent of the curve at the foot point. In other words, the bisector point must lie in the normal plane of the curve at the foot point. The last condition of Equation (10) implies that the bisector point Q must be at an equal distance from the two foot points.

When the two curves $C_1(t)$ and $C_2(r)$ are non-coplanar in \mathbb{R}^3 , the three vectors: $C_1'(t)$, $C_2'(r)$, and $C_1(t) - C_2(r)$ are linearly independent, in general. These three vectors form three rows in the following matrix of linear equations for the point $Q = (x, y, z)$ to satisfy:

$$\begin{bmatrix} C_1'(t) \\ C_2'(r) \\ C_1(t) - C_2(r) \end{bmatrix} \begin{bmatrix} x \\ y \\ z \end{bmatrix} = \begin{bmatrix} \langle C_1(t), C_1'(t) \rangle \\ \langle C_2(r), C_2'(r) \rangle \\ \frac{\|C_1(t)\|^2 - \|C_2(r)\|^2}{2} \end{bmatrix}.$$

Therefore, the point Q has a well defined solution for this equation. Using Cramer's rule, we can symbolically solve this equation for $Q = (x, y, z)$ and generate a rational bisector surface $Q(t, r)$ (see also Reference [4]).

In the case of two planar curves $C_1(t) = (x_1(t), y_1(t))$ and $C_2(r) = (x_2(r), y_2(r))$, the three vectors: $C_1'(t)$, $C_2'(r)$, and $C_1(t) - C_2(r)$ are always linearly dependent. When the two tangents $C_1'(t)$ and $C_2'(r)$ are neither parallel nor opposite, the point $P = (x, y)$ on the intersection of the two normal lines of the two curves has a unique symbolic solution for the following matrix equation:

$$\begin{bmatrix} C_1'(t) \\ C_2'(r) \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} \langle C_1(t), C_1'(t) \rangle \\ \langle C_2(r), C_2'(r) \rangle \end{bmatrix}. \quad (11)$$

Using Cramer's rule, we can generate a planar bivariate rational surface: $P(t, r) = (x(t, r), y(t, r))$, which is embedded in the xy -plane:

$$x(t, r) = \frac{\begin{vmatrix} x_1(t)x_1'(t) + y_1(t)y_1'(t) & y_1'(t) \\ x_2(r)x_2'(r) + y_2(r)y_2'(r) & y_2'(r) \end{vmatrix}}{\begin{vmatrix} x_1'(t) & y_1'(t) \\ x_2'(r) & y_2'(r) \end{vmatrix}}$$

$$y(t, r) = \frac{\begin{vmatrix} x_1'(t) & x_1(t)x_1'(t) + y_1(t)y_1'(t) \\ x_2'(r) & x_2(r)x_2'(r) + y_2(r)y_2'(r) \end{vmatrix}}{\begin{vmatrix} x_1'(t) & y_1'(t) \\ x_2'(r) & y_2'(r) \end{vmatrix}}. \quad (12)$$

Each surface point $P(t, r)$ is evaluated as the intersection point of the two normal lines of $C_1(t)$ and $C_2(r)$, respectively.

For a point $P(t, r)$ to be on the bisector curve, this point must satisfy the equidistance condition of Equation (10). By substituting the above rational solution $P(t, r) = (x(t, r), y(t, r))$ into Equation (10), we get the following bivariate rational function the zero-set of which corresponds to the bisector curve:

$$0 = \hat{\mathcal{F}}_3(t, r) \quad (13)$$

$$= \left\langle P(t, r) - \frac{C_1(t) + C_2(r)}{2}, C_1(t) - C_2(r) \right\rangle.$$

The denominator of $P(t, r)$ is $x_1'(t)y_2'(r) - x_2'(r)y_1'(t)$, which is the same as the redundant component of Equation (4). By multiplying this term to Equation (13), we get

$$0 = \mathcal{F}_3(t, r)$$

$$= (x_1'(t)y_2'(r) - x_2'(r)y_1'(t))\hat{\mathcal{F}}_3(t, r).$$

Clearly, $\mathcal{F}_3(t, r)$ is a polynomial function when both $C_1(t)$ and $C_2(r)$ are polynomial curves. Note that the term $x_1'(t)y_2'(r) - x_2'(r)y_1'(t)$ is not a factor of $\mathcal{F}_3(t, r)$ since it is the denominator of $\hat{\mathcal{F}}_3(t, r)$. However, $\mathcal{F}_1(t, r)$ and $\mathcal{F}_2(t, r)$ always contain this term as a redundant factor, which corresponds to the bisector points at infinity.

3.4 Degree Comparison

The bisector curve of $C_1(t)$ and $C_2(r)$ is a non-rational algebraic curve, in general [9]. For a planar point $P = (x, y)$ to be on the bisector of $C_1(t)$ and $C_2(r)$, this point must satisfy the three constraints of Equations (8)–(10). The conventional approach of Hoffmann and Vermeer [13] eliminates auxiliary variables t and r from the simultaneous system of polynomial equations. Then, the bisector curve is obtained as an implicit algebraic equation of x and y . The elimination process requires considerable memory space and produces an implicit algebraic equation of high degree. By contrast, and as we proposed here, it is an easier task to *match* the corresponding foot points on the two curves $C_1(t)$ and $C_2(r)$.

We have shown that the bivariate functions $\mathcal{F}_i(t, r)$, ($i = 1, 2, 3$), can be symbolically computed and represented as a polynomial or rational function. By employing only the numerator of each rational function $\mathcal{F}_i(t, r)$, all the matching points can be represented as the zero-set of a bivariate polynomial function of moderate degree. Given two planar polynomial curves $C_1(t)$ and $C_2(r)$ of degree m , the resulting function \mathcal{F}_i , ($i = 1, 2$), is a polynomial of degree:

$$2(m + (m - 1)) + 2(m - 1) = 6m - 4, \quad (14)$$

and \mathcal{F}_3 is a polynomial function of degree:

$$2m + 2(m - 1) = 4m - 2, \quad (15)$$

since the numerator and denominator of $P(t, r)$ are of polynomial degree $3m - 2$ and $2m - 2$, respectively (see Equation (12)). (Note that the difference in the degrees of Equations (14) and (15) is $2m - 2$, which is the degree of the redundant factor $x'_1(t)y'_2(r) - x'_2(r)y'_1(t)$ of $\mathcal{F}_1(t, r)$ and $\mathcal{F}_2(t, r)$.) For two cubic polynomial curves $C_1(t)$ and $C_2(r)$ (i.e., $m = 3$), the degree of \mathcal{F}_i , ($i = 1, 2$), is 14, and that of \mathcal{F}_3 is 10, which are moderate degrees to handle, in practice.

It would be quite interesting to compare the degree of $\mathcal{F}_3(s, t)$ with that of the corresponding bisector curve in the xy -plane. To the best of the authors' knowledge, there has been no previous result that can estimate the exact algebraic degree of such a bisector curve in the xy -plane. In the case of low degree curves $C_1(t)$ and $C_2(r)$ of degrees m_1 and m_2 , respectively, ($1 \leq m_1 \leq 3$, and $1 \leq m_2 \leq 6$), using Mathematica [17], we have observed that the bisector curve in the xy -plane is an algebraic curve of degree $7m_1m_2 - 3(m_1 + m_2) + 1$, which is also irreducible over rational coefficients. For two cubic curves, the degree is thus 46, which is considerably higher than the degree 10 of $\mathcal{F}_3(t, r)$. We also tried to compute the bisector curve (in the xy -plane) of higher degree curves (e.g., $m_1, m_2 \geq 4$). However, we have not been successful in getting a result due to the lack of memory space in computing Sylvester resultants employed in the variable elimination procedure. Consequently, memory efficiency also justifies our approach.

3.5 Computing Bisector Points

Let $C_1(t)$ and $C_2(r)$ be two matching foot points for a bisector point $P(t, r)$. Then the normal lines of $C_1(t)$ and $C_2(r)$ intersect at the bisector point $P(t, r)$. Each point in the zero-set of $\mathcal{F}_i(t, r) = 0$, $i = 1, 2, 3$, corresponds to a pair $(C_1(t), C_2(r))$ of matching foot points and thus to a bisector point $P(t, r) = (x(t, r), y(t, r))$ that can be computed by solving the following matrix equation:

$$\begin{bmatrix} C_1'(t) \\ C_2'(r) \end{bmatrix} \begin{bmatrix} x(t, r) \\ y(t, r) \end{bmatrix} = \begin{bmatrix} \langle C_1(t), C_1'(t) \rangle \\ \langle C_2(r), C_2'(r) \rangle \end{bmatrix}. \quad (16)$$

4 Experimental Results

The computation of the bivariate functions $\mathcal{F}_i(t, r)$ is quite simple and efficient. Using symbolic tools to compute the summation, difference and product of (piecewise) polynomial/rational forms [3], we can derive (piecewise) polynomial/rational functions representing the functions $\mathcal{F}_i(t, r)$. Figure 2 shows a simple example of the bisector computation for a line and a quadratic curve. The two separated segments in the zero-set of $\mathcal{F}_1(t, r)$ in Figure 2(b) are mapped back into the bisector curve segments in \mathbb{R}^2 as shown in Figure 2(a). Note that the i -th segment of the zero-set in Figure 2(b) corresponds to the i -th bisector curve segment, $\mathcal{C}_{12}^i(t, r)$, in Figure 2(a). We follow a similar convention in most other examples, too. Figure 3(a) shows the bisector curve of two quadratic Bézier curve segments. In Figures 3(b)–(c), the functions $\mathcal{F}_1(t, r)$ and $\mathcal{F}_3(t, r)$ are shown along with their zero-sets. In all figures of this paper, we show curve–curve bisectors only. Point–point and point–curve bisectors are easy to compute since they have closed-form representation [7].

Note that the fourth segment $\mathcal{C}_{12}^4(t, r)$ is not shown in Figure 3(a) since all (but one) points on this segments appear at infinity in the xy -plane (see the discussion below Equation (4)). The intersection point of the first and fourth components in Figure 3(b) generates a bisector point in the real, affine xy -plane, which is already in the first segment $\mathcal{C}_{12}^1(t, r)$. Therefore, the fourth component of Figure 3(b) is redundant.

The three major construction steps in computing a bisector curve are: (i) formulating a bivariate polynomial function $\mathcal{F}_i(t, r)$, (ii) computing the zero-set of $\mathcal{F}_i(t, r) = 0$, and (iii) generating bisector points from pairs $(C_1(t), C_2(r))$ of matching foot points. According to our experimental results, all three steps were found to be reasonably efficient. The symbolic computation of $\mathcal{F}_i(t, r)$ took from a fraction of a second to a few seconds on a high-end workstation for all the examples demonstrated in this paper. The zero-set finding of $\mathcal{F}_i(t, r) = 0$ and the generation of bisector points took from several seconds to a minute. All the experiments were carried out on a 200 Mhz R4400 SGI machine using the IRIT [6] modeling environment.

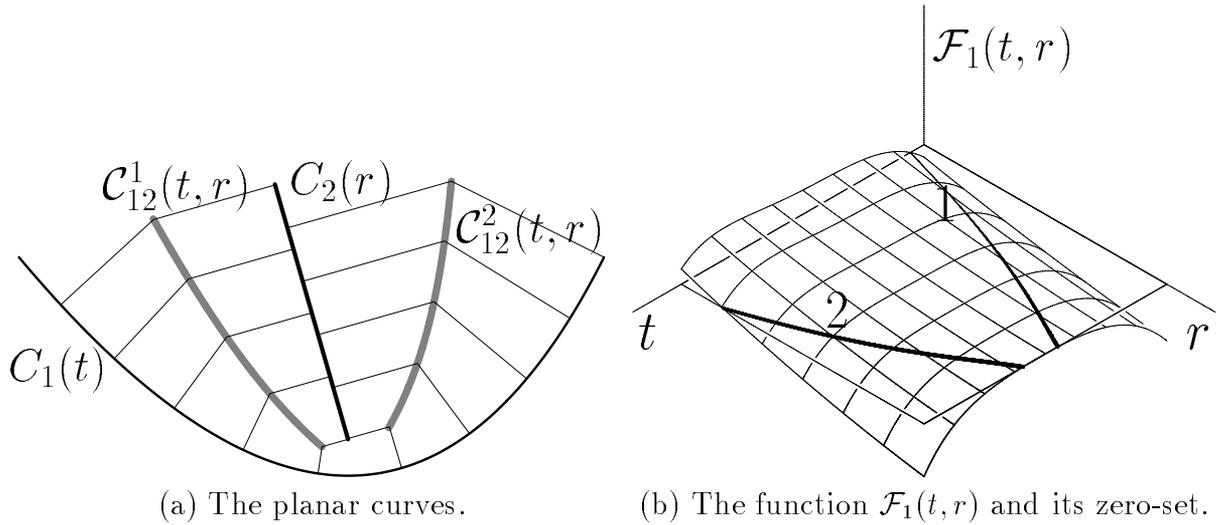


Figure 2: Bisector curve segments (in gray), $\mathcal{C}_{12}^i(t, r)$, ($i = 1, 2$), between a quadratic Bézier curve, $C_1(t)$, and a linear Bézier curve, $C_2(r)$. The i -th segment of the zero-set in (b) corresponds to $\mathcal{C}_{12}^i(t, r)$ in (a). In (a), thin lines are used to match the foot points; we follow a similar convention in most of the following figures of this paper. Note that only curve–curve bisectors are shown, purging point–point and point–curve bisectors.

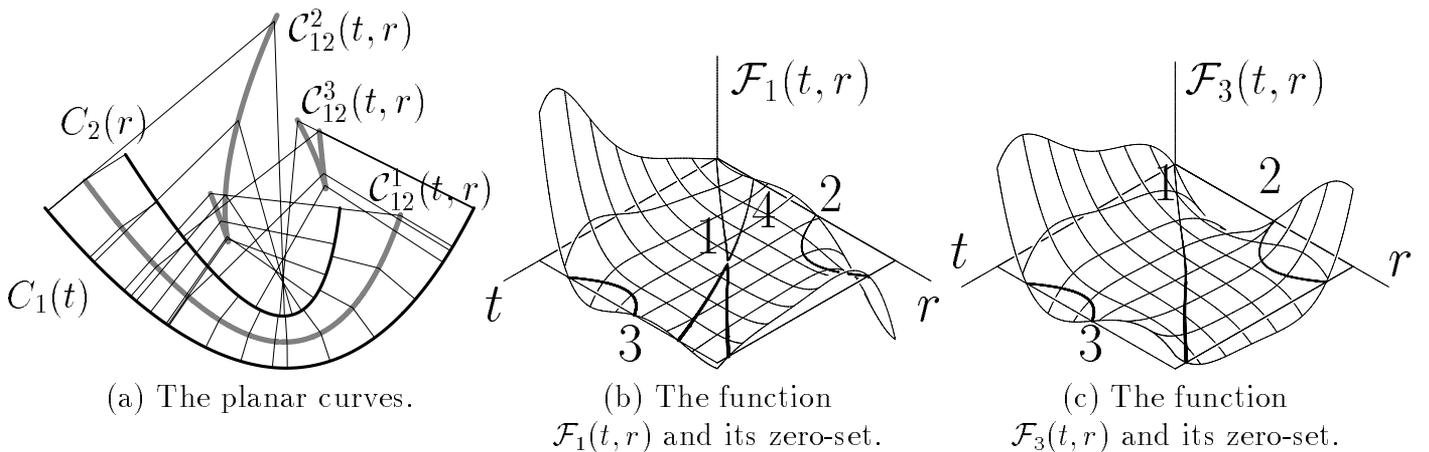


Figure 3: Bisector curve segments (in gray), $\mathcal{C}_{12}^i(t, r)$, ($i = 1, 2, 3, 4$), between two quadratic Bézier curves, $C_j(t)$, ($j = 1, 2$). The fourth segment $\mathcal{C}_{12}^4(t, r)$ is not shown in (a) since it is approaching infinity, being a bisector between points with colinear normals.

The zero-set finding is essentially a computational procedure that requires finding all the points along the intersection curve between the graph surface $\mathcal{S}_i(t, r) = (t, r, \mathcal{F}_i(t, r))$ and the tr -plane, which is a special case of the surface surface intersection (SSI) problem. Note that there are numerous research results for the SSI problem reported in the literature.

Among them, subdivision-based methods produce the most reliable and robust solutions, in general. They are usually slower than other sophisticated methods based on curve tracing. Quite often, other methods also use a preprocessing step that is based on subdivision. In this paper, due to robustness consideration, we used an adaptive subdivision-based scheme for finding the zero-set of $\mathcal{F}_i(t, r) = 0$. As we have observed above, using this method, less than one minute was required to construct each of the examples demonstrated in this paper. Therefore, the gain in robustness justifies our approach.

This robust adaptive subdivision approach depends on an ability to represent the bivariate functions $\mathcal{F}_i(t, r)$ symbolically. By searching for the extreme control points of each surface subregion during the subdivision, and exploiting the convex hull property of the Bézier and B-spline representations, we can efficiently extract the surface subregions that intersect with the tr -plane. When the remaining surface patches become sufficiently flat, we approximate these surface patches with polygons (in the $tr\mathcal{F}_i$ -space). The intersection of these polygons with the tr -plane provides a polygonal approximation of the zero-set of $\mathcal{F}_i(t, r) = 0$. We applied a numerical improvement procedure (based on local Newton-Raphson steps) to a piecewise linear approximation of the zero-set. Final results have very high precision, with typical tolerances of six orders of magnitude.

The degree of each function $\mathcal{F}_i(t, r)$ is higher than the two original curves. For two quadratic curves, $\mathcal{F}_1(t, r)$ and $\mathcal{F}_2(t, r)$ are of degree (8,8), whereas $\mathcal{F}_3(t, r)$ has degree (6,6). For two cubic curves, the degrees of $\mathcal{F}_1(t, r)$ and $\mathcal{F}_2(t, r)$ are (14,14), whereas $\mathcal{F}_3(t, r)$ has degree (10,10) (see Equations (14)–(15)). These degrees are not very high. In fact, we have experienced no practical limitation in computing the zero-sets of these functions.

It is interesting to note that we can also compute the bisector of a curve with itself, or the self-bisector. By using the same curve $C_1(t) \equiv C_2(r)$ in the formulation of $\mathcal{F}_i(t, r)$, we can compute the desired function, the zero-set of which provides the self-bisector curve. Figures 4 and 5 show two such examples for a quadratic Bézier curve and a cubic Bézier curve, respectively. Since $C_1(t)$ and $C_2(r)$ are, in fact, the same curve, the resulting functions are (anti)symmetric with respect to the diagonal, $t = r$, of the parametric domain of $\mathcal{F}_i(t, r)$. In addition, the diagonal itself is a valid, yet degenerated, solution for the bisector! This fact manifests itself as a wide almost horizontal domain near the diagonal: $t = r$. Clearly, the zero-set extracted along this diagonal should be ignored, taking into consideration only the solution *below* (or *above*) the diagonal $t = r$ due to the symmetry. Alternatively, $\mathcal{F}_i(t, r)$ can be divided by $(t-r)$ to the proper power to eliminate the artificial zero-set along the diagonal.

Finally, Figures 6 and 7 show more examples of the bisector for a cubic B-spline curve and a line, and the bisector of two quadratic curves, respectively.

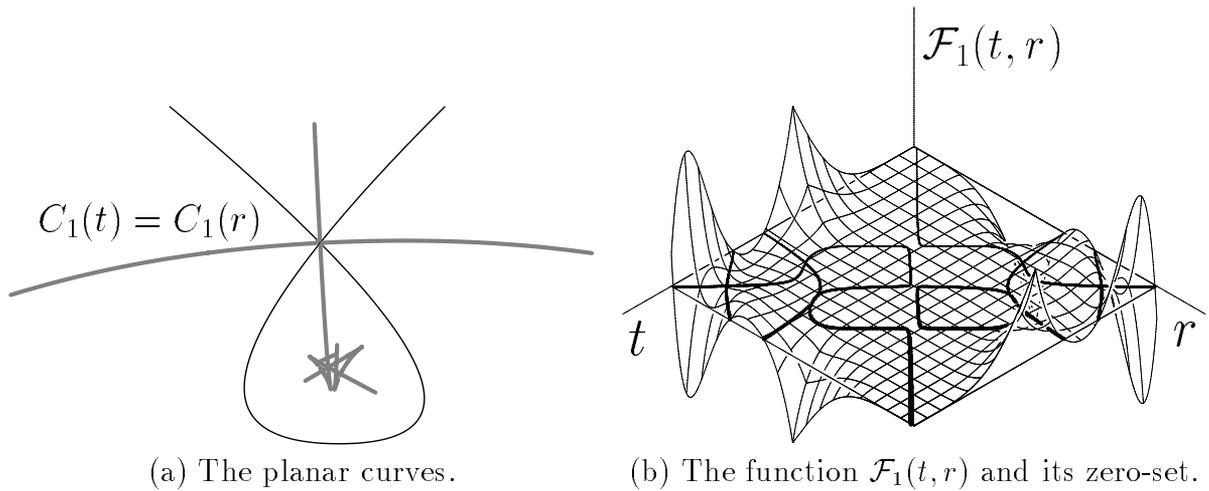


Figure 4: Self-bisector curve (in gray) of a quadratic B-spline curve with seven control points. Note the (anti)symmetry in $\mathcal{F}_1(t, r)$ with respect to the diagonal. Only the solution of $\mathcal{F}_1(t, r) = 0$ below the diagonal of $t = r$ needs to be considered.

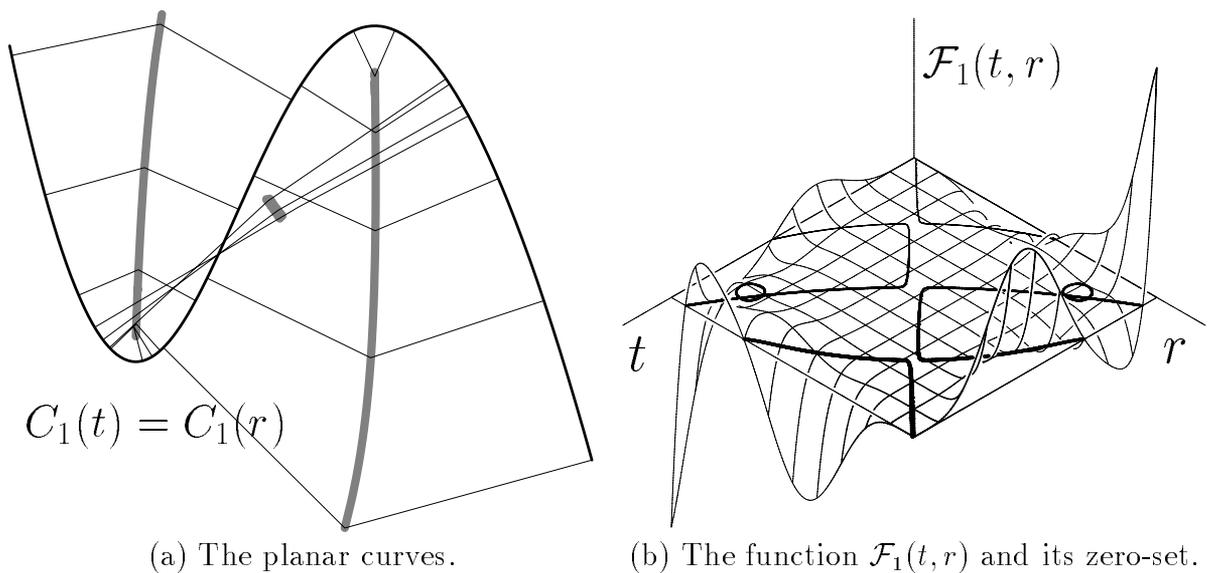


Figure 5: Self-bisector curve (in gray) of a cubic B-spline curve with seven control points. Note the (anti)symmetry in $\mathcal{F}_1(t, r)$ with respect to the diagonal. Only the solution of $\mathcal{F}_1(t, r) = 0$ below the diagonal of $t = r$ needs to be considered.

5 Conclusion

We have presented a new representation scheme for the bisector of two planar curves $C_1(t)$ and $C_2(r)$. The bisector curve is represented as a zero-set, $\mathcal{F}_i(s, t) = 0$, of a bivariate

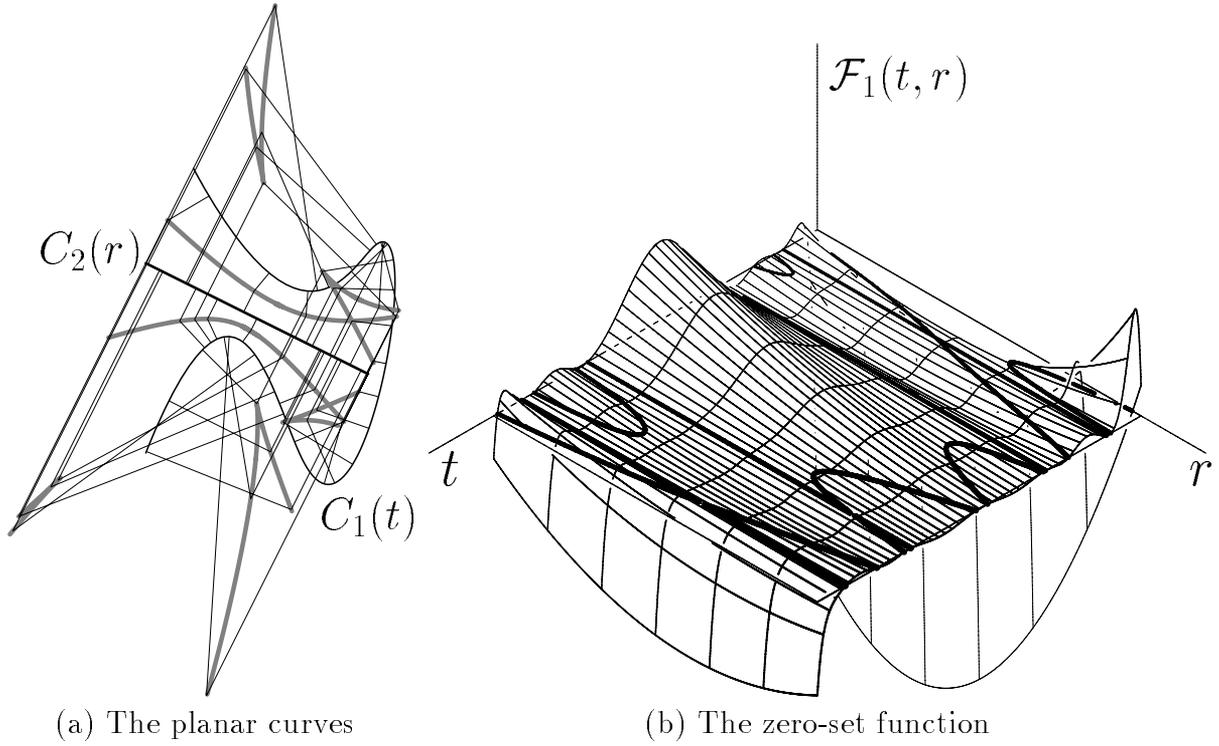


Figure 6: Bisector curve (in gray) between a cubic B-spline curve with eleven control points $C_1(t)$ and a linear curve $C_2(r)$.

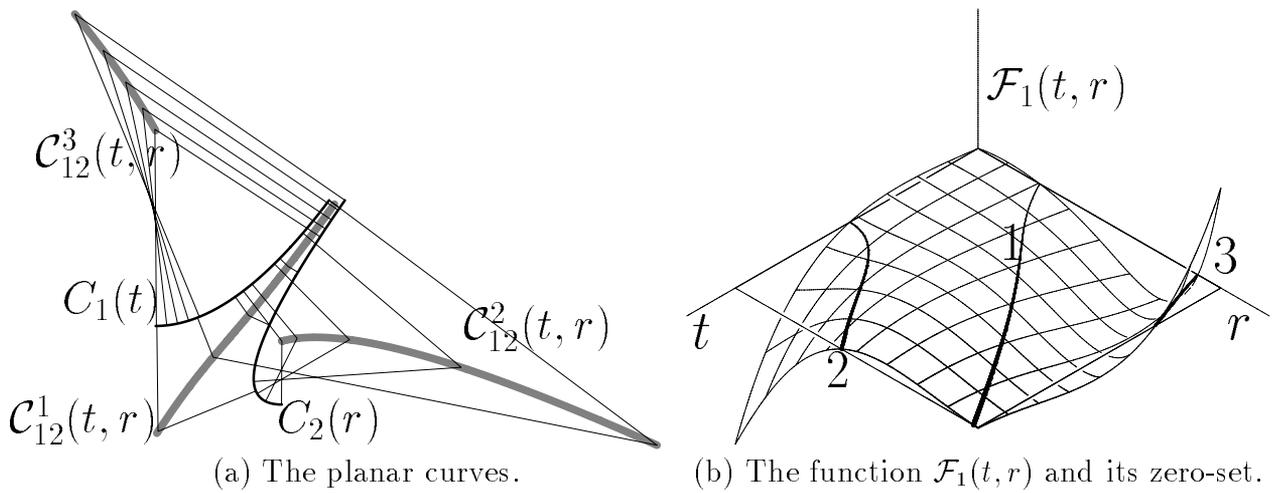


Figure 7: Bisector curve (in gray), $\mathcal{C}_{12}^i(t, r)$, ($i = 1, 2, 3$), between two quadratic Bézier curves, $C_j(t)$, ($j = 1, 2$).

polynomial function in t and r ; namely, as an implicit curve in the tr -plane (instead of a curve in the xy -plane). This tr -representation is considerably simpler and more efficient to

compute than the conventional representation scheme of bisector curves in the xy -plane [13].

Another important advantage of the approach proposed in this paper stems from the ability to apply the same basic strategy to bisector computation for higher dimensional varieties. We are currently investigating this generalization.

6 Acknowledgements

The authors would like to thank anonymous referees for their invaluable comments.

References

- [1] CHOI, J.-J. *Local Canonical Cubic Curve Tracing along Surface/Surface Intersections*. Ph.D. Thesis, Dept. of Computer Science, POSTECH, Pohang, Korea, February 1997.
- [2] CHOU, J. Voronoi diagrams for planar shapes. *IEEE Computer Graphics and Applications*, Vol. 15, No. 2, (1995), pp. 52–59.
- [3] ELBER, G., AND COHEN, E. Second order surface analysis using hybrid symbolic and numeric operators. *ACM Transactions on Graphics*, Vol. 12, No. 2, pp. 160–178, April 1993.
- [4] ELBER, G., AND KIM, M.-S.. The bisector surface of rational space curves. *ACM Transactions on Graphics*, Vol. 17, No. 1, pp. 32–49, 1998.
- [5] HEO, H.-S., KIM, M.-S., AND ELBER, G. Ruled/Ruled Surface Intersection. To appear in *Computer-Aided Design*, 1999.
- [6] ELBER, G. *IRIT 7.0a User's Manual*. Technion, 1996. <http://www.cs.technion.ac.il/~irit>.
- [7] FAROUKI, R., AND JOHNSTONE, J. The bisector of a point and plane parametric curve. *Computer Aided Geometric Design*, Vol. 11, No. 2, (1994), 117–151.
- [8] FAROUKI, R., AND JOHNSTONE, J. Computing point/curve and curve/curve bisectors. In *Design and Application of Curves and Surfaces: Mathematics of Surfaces V*, R. B. Fisher, Ed., Oxford University Press, (1994), pp. 327–354.
- [9] FAROUKI, R., AND RAMAMURTHY, R. Specified-precision computation of curve/curve bisectors. To appear in *International Journal of Computational Geometry & Applications*, (1997).

- [10] HELD, M. *On the Computational Geometry of Pocket Machining*, Lecture Notes in Computer Science, Vol. 500, Springer-Verlag, Berlin, (1991).
- [11] HOFFMANN, C. A dimensionality paradigm for surface interrogations. *Computer Aided Geometric Design*, Vol. 7, No. 6, (1990), pp. 517–532.
- [12] HOFFMANN, C. Computer vision, descriptive geometry, and classical mechanics. *Computer Graphics and Mathematics*, B. Falcidieno, I. Herman, and C. Pienovi (Eds.), Springer-Verlag, Berlin, 1992, pp. 229–243. Also available as Technical Report, CSD-TR-91-073, Computer Sciences Department, Purdue University, October, 1991.
- [13] HOFFMANN, C., AND VERMEER, P. Eliminating extraneous solutions in curve and surface operation. *Int'l J. of Computational Geometry and Applications*, Vol. 1, No. 1, (1991), pp. 47–66.
- [14] LAVENDER, D., BOWYER, A., DAVENPORT, J., WALLIS, A., AND WOODWARK, J. Voronoi diagrams of set-theoretic solid models. *IEEE Computer Graphics and Applications*, Vol. 12, No. 5, (1992), pp. 69–77.
- [15] PERSSON, H. NC machining of arbitrarily shaped pockets. *Computer-Aided Design*, Vol. 10, No. 3, (1978), pp. 169–174.
- [16] RAMAMURTHY, R. *Voronoi Diagrams and Medial Axes of Planar Domains with Curved Boundaries*. Ph.D. Thesis, Dept. of Mechanical Engineering, The Univ. of Michigan, Ann Arbor, USA, 1998.
- [17] S. WOLFRAM *Mathematica*, 2nd Ed., Addison-Wesley, (1991).