# A Computational Model for Nonrational Bisector Surfaces: Curve-Surface and Surface-Surface Bisectors

*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*
Dept. of Computer Engineering
Seoul National University
Seoul 151-742, Korea
E-mail: mskim@comp.snu.ac.kr

**Abstract**

The bisector of two rational surfaces in $\mathbb{R}^3$ is nonrational, in general. Moreover, in $\mathbb{R}^3$, the bisector of a rational curve and a rational surface is also nonrational. Thus, the bisector surfaces in these two special cases must be approximated numerically. Unfortunately, these bisector surfaces are algebraic surfaces of very high degree; thus the numerical approximation is a non-trivial task. This paper suggests a new computational model for constructing the curve-surface and surface-surface bisectors. The curve-surface bisector problem is reformulated as a trivariate zero-set finding problem, whereas the surface-surface bisector problem is reduced to that of finding the common zero-set of two four-variate functions.

## 1 Introduction

The medial axis of a 3D object consists of various bisector surfaces: point-point, point-curve, point-surface, curve-curve, curve-surface, and surface-surface bisectors. The point-point bisector is the plane of symmetry for two input points. The point-curve bisector is a rational developable surface when the input curve is rational [3, 10]. Moreover, the point-surface bisector is a rational surface for a rational input surface [6]. The bisector of two rational space curves is also a rational surface [3]. Unfortunately, the curve-surface and surface-surface bisectors are in general nonrational even if the input curves and surfaces are all rational. They are algebraic surfaces of very high degree. Thus, even numerical approximation to these high degree surfaces is a non-trivial task. Given two rational curves in the plane, their bisector is a nonrational algebraic curve. Farouki and Ramamurthy [8, 11] present a robust numerical tracing algorithm for the bisector curve; however, it is not easy to extend this approach to a similar problem in higher dimensions.

This paper suggests an alternative way of representing the nonrational bisector surfaces. The geometric contraints for bisector surfaces are represented in the $uvt$ or $uvst$-space, where $u, v, s, t$ are the parameters of input curves and surfaces. An important advantage of this alternative representation is that the degrees of constraint equations in the $uvt$ or $uvst$-space are considerably lower than those of bisector surfaces in the $xyz$-space. The approach of this paper is similar to that of Elber and Kim [4].

Given two planar rational curves $C_1(u)$ and $C_2(v)$ of degrees $d_1$ and $d_2$, respectively, Elber and Kim [4] reformulate the curve-curve bisector problem as a zero-set finding problem for a polynomial equation of degree $2(d_1 + d_2) - 2$ in the $uv$-plane. For example, given two cubic curves in the plane, we need to solve a polynomial equation of degree 10. This is a considerably lower degree than that of the bisector curve in the $xy$-plane as discussed below.

Though there has been no formal proof developed for arbitrary degrees, using Mathematica [12], we observed that the bisector curve in the $xy$-plane has an algebraic degree of $7d_1d_2 - 3(d_1 + d_2) + 1$, for $1 \leq d_1 \leq 3$ and $1 \leq d_2 \leq 6$. (Because of the lack of memory space, we have not been successful in checking the degree analysis for $d_1, d_2 \geq 4$.) This means that the bisector of two cubic polynomial curves has degree 46, which is too high to be useful in practice. Thus, this rather preliminary degree analysis justifies the approach of Elber and Kim [4].

In the case of the curve-surface and surface-surface bisectors, we will end up with algebraic bisector surfaces of even higher degree in the $xyz$-space. Because of this limitation, previous work on bisector surfaces has focused on simple surfaces of special types and configurations [2, 5, 6]. In this paper, we take a first step for computing the curve-surface and surface-surface bisectors for arbitrary rational curves and surfaces.

We represent the geometric constraints for the curve-surface bisector using a polynomial equation. The polynomial equation is given in terms of the parameters of the input curve and surface. Thus, the bisector constraints are represented as the zero-set of a trivariate polynomial function, i.e., the zero-set is given as an algebraic surface in the $uvt$-space. The curve-surface bisector is then constructed by transforming the zero-set into the $xyz$-space using a rational map. Moreover, the surface-surface bisector is represented by the common zero-set of two four-variate polynomial equations. The two equations are given in the parameters of two input surfaces; that is, the common zero-set is given as the intersection of two algebraic hypersurfaces in the $uvst$-space. There is a rational map from the solution space to the bisector surface; thus, the construction of bisector surfaces is easy once we construct the common zero-set of two polynomial equations in four variables.

The rest of this paper is organized as follows. Section 2 reformulates the curve-surface and surface-surface bisector problems as zero-set finding problems. In Section 3, we present numerical methods for computing the zero-sets. In Section 4, we present some experimental results that demonstrate the effectiveness of our approach. We conclude this paper with some final remarks in Section 5.

# 2 Computational Model

In this section, we propose a computational model for the curve-surface and surface-surface bisectors in $\mathbb{R}^3$. We consider the bisector surface of a rational space curve and a rational surface in Section 2.1, and that of two rational surfaces in Section 2.2.

## 2.1 Curve–Surface Bisector

Let $C(t)$ and $S(u,v)$ be regular rational curve and surface, respectively. We assume that they are sufficiently differentiable as needed. Consider a bisector point $\mathcal{B}$ between $S(u,v)$ and $C(t)$:

$$0 = \left\langle \mathcal{B} - S(u,v), \frac{\partial S(u,v)}{\partial u} \right\rangle, \tag{1}$$

$$0 = \left\langle \mathcal{B} - S(u,v), \frac{\partial S(u,v)}{\partial v} \right\rangle, \tag{2}$$

$$0 = \left\langle \mathcal{B} - C(t), \frac{dC(t)}{dt} \right\rangle, \tag{3}$$

$$0 = \langle \mathcal{B} - C(t), \mathcal{B} - C(t) \rangle - \langle \mathcal{B} - S(u,v), \mathcal{B} - S(u,v) \rangle. \tag{4}$$

Equations (1)–(2) mean that the bisector point $\mathcal{B}$ is located on the normal line of $S(u,v)$, whereas Equation (3) implies that $\mathcal{B}$ is contained in the normal plane of $C(t)$. Moreover, Equation (4) constrains the bisector point $\mathcal{B}$ to the symmetry plane of $S(u,v)$ and $C(t)$:

$$2 \langle \mathcal{B}, S(u,v) - C(t) \rangle + \|C(t)\|^2 - \|S(u,v)\|^2 = 0, \tag{5}$$

which is obtained by cancelling out the square term $\|\mathcal{B}\|^2$ from Equation (4). Note that this symmetry plane contains all points equidistant from the two points $S(u,v)$ and $C(t)$.

Equations (1)–(4) are all linear in $\mathcal{B}$. These equations would produce a rational representation for the bisector $\mathcal{B}$ if the problem were given in $\mathbb{R}^n$, for $n \geq 4$. Unfortunately, we are restricted to $\mathbb{R}^3$ which does not provide sufficient degrees of freedom. Thus, the curve-surface bisector in $\mathbb{R}^3$ is in general a nonrational algebraic surface, the computation of which is not easy.

Given a curve and a surface in general position, consider the intersection between the normal line of the surface $S(u,v)$ and the normal plane of the curve $C(t)$. This point is given as the solution of the first three Equations (1)–(3). The point of intersection $\mathcal{B}(u,v,t)$ is given a rational representation as a trivariate volume in $\mathbb{R}^3$. When we substitute this trivariate representation into Equation (5), we get the following constraint equation:

$$2 \langle \mathcal{B}(u,v,t), S(u,v) - C(t) \rangle + \|C(t)\|^2 - \|S(u,v)\|^2 = 0. \tag{6}$$

When $S(u,v)$ and $C(t)$ are both rational, the above equation represents an algebraic surface in the $uvt$-space.

A rational representation of $\mathcal{B}(u,v,t)$ is obtained by solving Equations (1)–(3). The solution based on Cramer's rule involves the determinants of $3 \times 3$ matrices. We take a direct geometric approach that

derives a closed-form solution for the intersection point. The intersection point must be located on the normal line of the surface $S(u, v)$:

$$\mathcal{B} = S(u, v) + \alpha n(u, v), \tag{7}$$

where $n(u, v) = \frac{\partial S(u,v)}{\partial u} \times \frac{\partial S(u,v)}{\partial v}$ is an unnormalized normal vector of $S(u, v)$, and $\alpha \in I\!R$. Equations (1)–(2) are now clearly satisfied. Then, by substituting $\mathcal{B}$ into Equation (3), we get

$$0 = \left\langle S(u, v) + \alpha n(u, v) - C(t), \frac{dC(t)}{dt} \right\rangle,$$

or equivalently,

$$\alpha = \alpha(u, v, t) = \frac{\left\langle C(t) - S(u, v), \frac{dC(t)}{dt} \right\rangle}{\left\langle n(u, v), \frac{dC(t)}{dt} \right\rangle}. \tag{8}$$

The point $\mathcal{B}(u, v, t) = S(u, v) + \alpha(u, v, t)n(u, v)$ is the intersection point between the normal line of the surface and the normal plane of the curve. By substituting this point of intersection into Equation (4) or Equation (5), meaning an equal distance from $S(u, v)$ and $C(t)$, we get the following constraint equation:

$$\begin{aligned} 0 &= \langle S(u, v) - C(t) + \alpha(u, v, t)n(u, v), S(u, v) - C(t) + \alpha(u, v, t)n(u, v)\rangle \\ &\quad - \langle \alpha(u, v, t)n(u, v), \alpha(u, v, t)n(u, v)\rangle . \\ &= \|S(u, v) - C(t)\|^2 + \alpha(u, v, t) \langle n(u, v), S(u, v) - C(t)\rangle \end{aligned} \tag{9}$$

The zero-set of Equation (9) satisfies all four constraints of Equations (1)–(4). The first three constraints are satisfied via the proper selection of $\alpha(u, v, t)$ in Equation (8), whereas the fourth constraint is satisfied by the substitution of $\alpha(u, v, t)$ into the constraint, as given in Equation (9).

Consquently, we have reduced the problem of constructing a curve-surface bisector in $I\!R^3$ into a zero-set finding for a trivariate function as shown in Equation (9). In Section 3, we consider a numerical method for computing the zero-set of the trivariate function. For each point $(u, v, t)$ in the zero-set, the corresponding bisector point is computed by the following rational map:

$$\mathcal{B}(u, v, t) = S(u, v) + \frac{\left\langle C(t) - S(u, v), \frac{dC(t)}{dt} \right\rangle}{\left\langle n(u, v), \frac{dC(t)}{dt} \right\rangle} n(u, v). \tag{10}$$

### 2.2 Surface–Surface bisector

Let $S_1(u, v)$ and $S_2(s, t)$ be two regular rational surfaces which are sufficiently differentiable as needed. We consider the bisector surface $\mathcal{B}$ of $S_1(u, v)$ and $S_2(s, t)$ that satisfies the following five constraint equations:

$$0 = \left\langle \mathcal{B} - S_1(u, v), \frac{\partial S_1(u, v)}{\partial u} \right\rangle, \tag{11}$$

$$0 = \left\langle \mathcal{B} - S_1(u, v), \frac{\partial S_1(u, v)}{\partial v} \right\rangle, \tag{12}$$

$$0 = \left\langle \mathcal{B} - S_2(s, t), \frac{\partial S_2(s, t)}{\partial s} \right\rangle, \tag{13}$$

$$0 = \left\langle \mathcal{B} - S_2(s, t), \frac{\partial S_2(s, t)}{\partial t} \right\rangle, \tag{14}$$

$$0 = \langle \mathcal{B} - S_1(u, v), \mathcal{B} - S_1(u, v)\rangle - \langle \mathcal{B} - S_2(s, t), \mathcal{B} - S_2(s, t)\rangle . \tag{15}$$

Equations (11)–(12) mean that the bisector point $\mathcal{B}$ is located on the normal line of $S_1(u,v)$, whereas Equations (13)–(14) imply that $\mathcal{B}$ is on the normal line of $S_2(s,t)$. Moreover, Equation (15) constrains the point $\mathcal{B}$ to the symmetry plane of $S_1(u,v)$ and $S_2(s,t)$.

Equations (11)–(15) are all linear in $\mathcal{B}$. If the problem were given in $\mathbb{R}^n$, for $n \geq 5$, these five equations would produce a rational representation for the surface-surface bisector. However, the problem is restricted to $\mathbb{R}^3$. Thus, the computation of the surface-surface bisector in $\mathbb{R}^3$ is not easy.

Let $\mathcal{B} = S_1(u,v) + \alpha n_1(u,v)$, $\alpha \in \mathbb{R}$, be a bisector point located on the normal line of $S_1(u,v)$ (see Equation (7)), where $n_1(u,v) = \frac{\partial S_1(u,v)}{\partial u} \times \frac{\partial S_1(u,v)}{\partial v}$ is an unnormalized normal vector of $S_1(u,v)$. Then, Equations (11)–(12) are automatically satisfied. By substituting $\mathcal{B}$ into Equation (15), we get the following equation

$$0 = \langle \alpha n_1(u,v), \alpha n_1(u,v) \rangle - \langle S_1(u,v) + \alpha n_1(u,v) - S_2(s,t), S_1(u,v) + \alpha n_1(u,v) - S_2(s,t) \rangle$$

or equivalently,

$$\alpha(u,v,s,t) = \frac{\langle S_1(u,v) - S_2(s,t), S_1(u,v) - S_2(s,t) \rangle}{-2 \langle n_1(u,v), S_1(u,v) - S_2(s,t) \rangle}. \tag{16}$$

Finally, we substitute this representation of $\alpha(u,v,s,t)$ into $\mathcal{B} = S_1(u,v) + \alpha n_1(u,v)$, and update Equations (13)–(14) with this new representation of $\mathcal{B}$. Let $\Delta(u,v,s,t)$ be defined as follows:

$$\begin{aligned} \Delta(u,v,s,t) \;\; = \;\; & -2(S_1(u,v) - S_2(s,t)) \langle n_1(u,v), S_1(u,v) - S_2(s,t) \rangle \\ & + n_1(u,v) \langle S_1(u,v) - S_2(s,t), S_1(u,v) - S_2(s,t) \rangle . \end{aligned}$$

Then, Equations (13)–(14) reduce to

$$0 = \left\langle \Delta(u,v,s,t), \frac{\partial S_2(s,t)}{\partial s} \right\rangle, \tag{17}$$

$$0 = \left\langle \Delta(u,v,s,t), \frac{\partial S_2(s,t)}{\partial t} \right\rangle. \tag{18}$$

The common zero-set of the two Equations (17)–(18) satisfies all five constraints of Equations (11)–(15). Hence, we reduce the surface-surface bisector problem of $\mathbb{R}^3$ into a common zero-set finding problem for two four-variate functions of Equations (17)–(18). In Section 3, we consider numerical methods for computing this zero-set. For each point $(u,v,s,t)$ in the zero-set, the corresponding bisector point is computed by the following rational map:

$$\mathcal{B}(u,v,s,t) = S_1(u,v) - \frac{\langle S_1(u,v) - S_2(s,t), S_1(u,v) - S_2(s,t) \rangle}{2 \langle n_1(u,v), S_1(u,v) - S_2(s,t) \rangle} \; n_1(u,v). \tag{19}$$

## 3 Constructing the Bisector Surfaces

We now consider the problem of zero-set finding for multi-variate functions. In particular, we are interested in computing the zero-set of the trivariate function introduced in Section 2.1, and the common zero-set for the pair of four-variate functions introduced in Section 2.2. Both cases have two degrees of freedom left;

thus, the bisectors are surfaces in $\mathbb{R}^3$. The surfaces are in general nonrational algebraic surfaces; therefore they must be approximated numerically.

Taking a robust approach, we subdivide the multi-variate functions(s) until the function values are bounded within a certain resolution. Section 3.1 discusses this subdivision stage, which ends up with a set of discrete points. These discrete points approximate the zero-set. We then apply a numerical improvement to these discrete points, which is explained in Section 3.2. Finally, we fit a bivariate surface through the set of points sampled on the bisector surface. The resulting surface approximates the curve–surface and surface–surface bisectors. This surface fitting stage is discussed in Section 3.3

### 3.1 Subdivision Stage for Zero-Set Finding

Consider $n$ multi-variate polynomial functions $\mathcal{F}_i(u_1, u_2, \cdots, u_{m-1}) = 0$, $(i = 1, \cdots, n)$, in $\mathbb{R}^m$, where $n \leq m$. Denote by $\boldsymbol{u} = (u_1, u_2, \cdots, u_{m-1})$ a point in the $(m-1)$-dimensional parameter space of $\mathcal{F}_i$. We consider all simultaneous solution points, $\boldsymbol{u}^s \in \mathbb{R}^{m-1}$, such that $\mathcal{F}_i(\boldsymbol{u}^s) = 0$, for all $i = 1, \cdots, n$.

Assume that $\mathcal{F}_i$, $i = 1, \cdots, n$, are represented as B-spline or Bézier multi-variate scalar surfaces. Then, the domain of $\mathcal{F}_i(\mathbf{u})$ contains zeros only if the control coefficients of $\mathcal{F}_i$ have different signs; this is because the value of zero must be contained in the convex hull of the control coefficients of $\mathcal{F}_i$. Using this convex hull property, a subdivision-based approach as described in Algorithm 3.1 generates a set of discrete points that robustly converges to the simultaneous zero-set of $\mathcal{F}_i$, $i = 1, \cdots, n$.

Let $\tau$ be a subdivision tolerance that determines the maximum number of points to be generated in the subdivision stage. Assume that we subdivide the parameter domain of a unit hypercube of dimension $m-1$, while each axis direction being subdivided into $\approx \frac{1}{\tau}$ segments. The result will be a total of $\left(\frac{1}{\tau}\right)^{m-1}$ cells. Each such cell generates one point (i.e., the center of each cell); thus, we will end up with at most $\left(\frac{1}{\tau}\right)^{m-1}$ solution points at the end of this subdivision stage.

### 3.2 Numerical Improvement Step for Zero-Set Finding

This section discusses the numerical improvement for the approximated solutions. While essentially a Newton-Raphson step is described, we take a more geometrically intuitive approach.

Let $\boldsymbol{u}^0 = (u_1^0, u_2^0, \cdots, u_{m-1}^0)$ be an approximate solution as computed in Section 3.1. That is, $\mathcal{F}_i(\boldsymbol{u}^0) \approx 0$, for $i = 1, \cdots, n$. The normal space of $\mathcal{F}_i$ is a line in $\mathbb{R}^m$, and at $\boldsymbol{u}^0$ this line is parallel to

$$\boldsymbol{n}_i(\boldsymbol{u}^0) = \left(\frac{\partial \mathcal{F}_i}{\partial u_1}(\boldsymbol{u}^0), \frac{\partial \mathcal{F}_i}{\partial u_2}(\boldsymbol{u}^0), , \cdots, \frac{\partial \mathcal{F}_i}{\partial u_{m-1}}(\boldsymbol{u}^0), -1\right)$$

where $\frac{\partial \mathcal{F}_i}{\partial u_j}(\boldsymbol{u}^0)$ denotes the partial derivative of $\mathcal{F}_i$ with respect to $u_j$ evaluated at $\boldsymbol{u}^0$. Complementing the normal's one dimensional space at $\boldsymbol{u}^0$ is the tangent hyperplane of $\mathcal{F}_i(u_1, u_2, \cdots, u_{m-1})$, an $(m-1)$-variety function with parameters $\boldsymbol{u} = (u_1, u_2, \cdots, u_{m-1})$, that is equal to

$$\mathcal{P}_i(\boldsymbol{u}): \ \frac{\partial \mathcal{F}_i}{\partial u_1}(\boldsymbol{u}^0)u_1 + \frac{\partial \mathcal{F}_i}{\partial u_2}(\boldsymbol{u}^0)u_2 + \cdots + \frac{\partial \mathcal{F}_i}{\partial u_{m-1}}(\boldsymbol{u}^0)u_{m-1} + -u_m - \left\langle \boldsymbol{n}_i(\boldsymbol{u}^0), \overline{\boldsymbol{u}^0} \right\rangle = 0.$$

where $\overline{\boldsymbol{u}^0} = (u_1^0, u_2^0, \cdots, u_{m-1}^0, \mathcal{F}_i(\boldsymbol{u}^0))$.

**Algorithm 3.1**
**Input:**
$\mathcal{F}_i$, $i = 1, \cdots, n$, $n$ multivariate explicit constraints;
$\tau$, Tolerance of subdivision process;
**Output:**
$\mathcal{S}$, A set of points in the parametric space of $\mathcal{F}_i$ such that each point $\mathbf{u^s} \in \mathcal{S}$ yields an approximated simultaneous zero over all $\mathcal{F}_i$;
**ZeroSetSubdiv**( $\mathcal{F}_i$, $i = 1, \cdots, n$, $\left\{(u_j^{min}, u_j^{max})\right\}$, $j = 1, \cdots, (m-1)$, $\tau$ )
**begin**

$\Delta^{max} u \Leftarrow \max_j \left(u_j^{max} - u_j^{min}\right)$;
**if** $(\Delta^{max} u < \tau)$ **then begin**
**return** $\left\{\left(\frac{u_1^{min}+u_1^{max}}{2}, \frac{u_2^{min}+u_2^{max}}{2}, \cdots, \frac{u_{m-1}^{min}+u_{m-1}^{max}}{2}\right)\right\}$;
**end**
**else begin**
$p_i^{min}$, $p_i^{max} \Leftarrow$ Minimal and maximal control coefficients of $\mathcal{F}_i$;
**if** ($\exists i$ such that $p_i^{min} p_i^{max} > 0$) **then**
**return** $\phi$;
**else begin**
$\mathcal{F}_i^1$, $\mathcal{F}_i^2 \Leftarrow \mathcal{F}_i$ subdivided at $\frac{u_j^{min}+u_j^{max}}{2}$, along the $j$th parametric direction, where $\Delta^{max} u$ has been detected, $i = 1, \cdots, n$;
**return ZeroSetSubdiv**( $\mathcal{F}_i^1$, $i = 1, \cdots, n$,
$\left\{(u_0^{min}, u_0^{max}), \cdots (u_j^{min}, \frac{u_j^{min}+u_j^{max}}{2}), \cdots (u_n^{min}, u_n^{max}),\right\}$ ) $\cup$
**ZeroSetSubdiv**( $\mathcal{F}_i^2$, $i = 1, \cdots, n$,
$\left\{(u_0^{min}, u_0^{max}), \cdots (\frac{u_j^{min}+u_j^{max}}{2}, u_j^{max}), \cdots (u_n^{min}, u_n^{max}),\right\}$ );
**end**
**end**

Using a first order approximation of $\mathcal{F}_i$, each $\mathcal{F}_i$ function constrains the solution to be on the hyperplane $\mathcal{P}_i(\boldsymbol{u}_0)$. Noting that the planes $\mathcal{P}_i(\boldsymbol{u}_0)$ also contain $u_m$ as a free variable and recalling the need for $\mathcal{F}_i = 0$, $\forall i$, one can embed this requirement simply as an additional constraint of $u_m = 0$. Hence, one ends up with $n + 1$ linear constraints.

Consider the case of $m = n + 1$. One now has a fully constrained system of $m$ linear equations that are formulated from the $n$ tangent planes, $\mathcal{P}_i(\boldsymbol{u}^0)$, of the different $\mathcal{F}_i(\boldsymbol{u})$ functions, and one final constraint that is derived from the zero that is sought: $u_m = 0$. Therefore, we are having $m$ linear equations and $m$ unknowns. Note that $m$ planes in general position in $\mathbb{R}^m$ intersect at a single point, which is also the result of one iteration of a multidimensional Newton Raphson procedure in $\mathbb{R}^m$.

For example, consider the case of two scalar bivariate constraints in $\mathbb{R}^3$, $\mathcal{F}_i(x, y) = 0$, $i = 1, 2$. Consider the constraint of $z = 0$ together with $\mathcal{P}_i(x, y) = 0$, $i = 1, 2$. We then have three linear constraints and

three unknowns. Three planes in general position in $\mathbb{R}^3$ intersect at a single point, which is also the result of the next iteration of the Newton Raphson procedure.

Now let $m \leq n$. The linear system of equations is under-determined. Note that $n$ planes in $\mathbb{R}^m$ intersect in an $(m-n)$-dimensional variety, $\mathcal{V}$. One must employ a linear solver that yields a minimal $L_2$ norm solution, in order to find the closest solution to the current position, $\boldsymbol{u}^0$, out of the possible solution set, $\mathcal{V}$. SVD decomposition [1] and/or QR factorizations [1] are two options for solving this under-determined problem, producing a minimal $L_2$ norm solution.

In this paper, a QR factorization procedure has been employed. The tolerance employed in the numeric stage is several magnitudes smaller than that of the subdivision stage. Thus the vast majority of points were successfully improved to that tolerance in very few iterations, gaining more than an order of magnitude per iteration. Since we will use a surface fitting in the next step, we don't need to enforce all points from the subdivision stage to be improved. We can simply purge away a few points that fail to improve. Examples are presented in Section 4.

### 3.3 Surface Fitting to Bisector Surface

Once a set of points that satisfy the simultaneous zeros is constructed, we map these points onto the bisector surface. After that, we fit a surface to the set of discrete points on the bisector surface. In contrast to the problem of regular scattered data interpolation, our problem here allows a simple solution to the parameterization of data points.

The set of points in the simultaneous zero-set provides the correspondence from one surface to another curve or surface. Thus, we can employ the parameterization of the first surface as the parameterization of the bisector surface. This approach has been taken in all the examples that are presented in Section 4.

## 4 Experimental Results

In this section, we demonstrate the effectiveness of our approach by presenting several examples of curve-surface and surface-surface bisectors in $\mathbb{R}^3$ that are computed using the algorithm presented in this paper. Each example shows a set of discrete points on the bisector surface. The points are the result of the simultaneous solution of the multivariate constraint equations. An approximate bisector surface (shown in gray color) is constructed by interpolating the set of discrete points on the bisector surface.

Figure 1 shows two simple examples of curve-surface bisectors. A line-plane bisector is shown in Figure 1(a), and a curve-plane bisector is shown in Figure 1(b). Figure 2 shows a more complex bisector surface between a freeform curve and a convex surface. Figure 3 presents two surface-surface bisectors. The bisector surface between a plane and a convex surface is shown in Figure 3(a), and the bisector between a plane and a concave surface is shown in Figure 3(b). Note that the bisector surface of Figure 3(b) has a self-intersection. Finally, Figure 4 shows a bisector surface between two freeform rational surfaces.

The computing time for each of the above examples took from several seconds to a few minutes on a 150 MHz R5000 Indy machine. The most time-consuming step in this procedure is the generation of

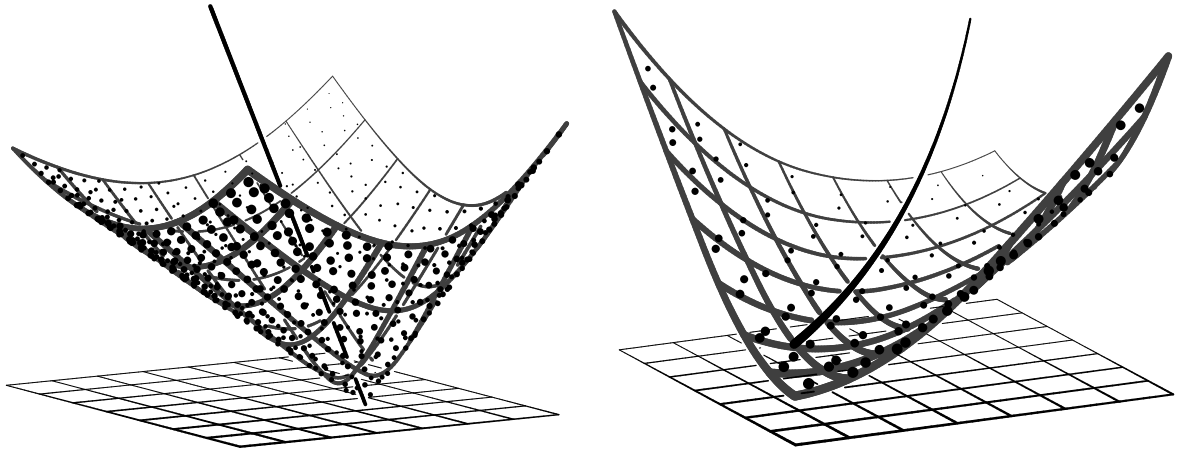



Figure 1: (a) Bisector surface (in gray) of a (tilted) line and a plane; and (b) bisector surface (in gray) of a (tilted) quadratic curve and a plane.

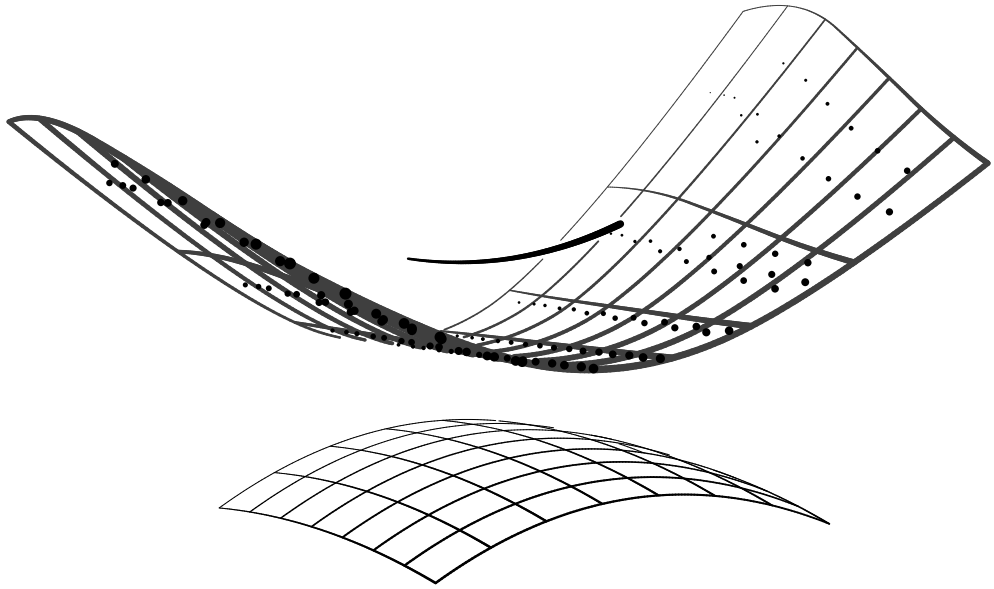
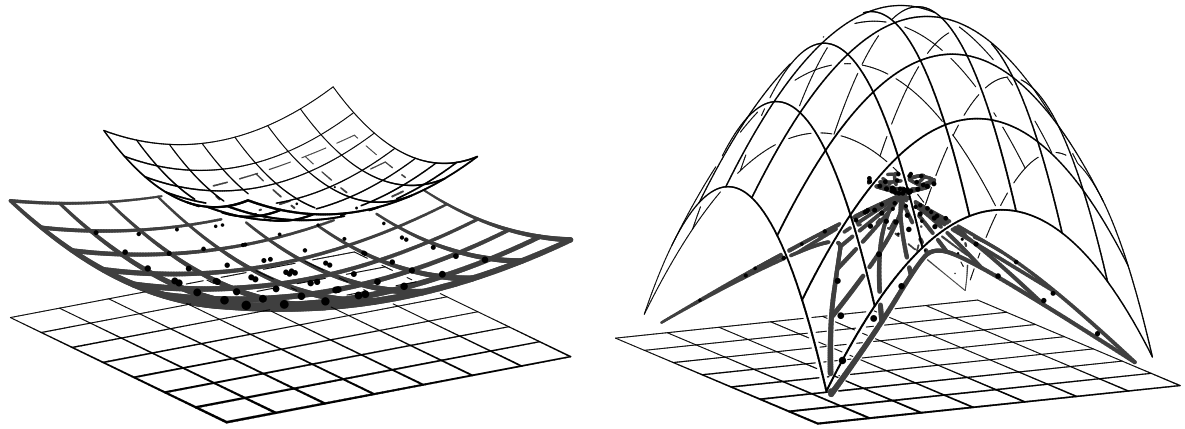


Figure 2: Bisector surface (in gray) of a quadratic curve and a convex surface.

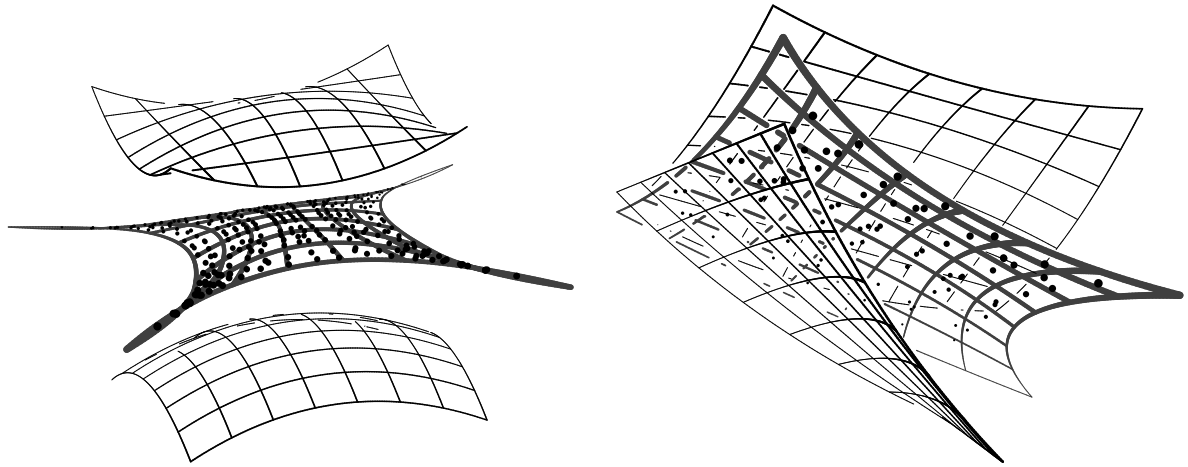
the set of discrete points that approximates the zero-set. The number of discrete points generated for the zero-set of each example is also listed in Table 1.

In all the examples presented so far, there is a one-to-one correspondence between the zero-set and an input surface. Thus, the mapping from an input surface to the other surface or even to the bisector surface is well-defined in each example. Nevertheless, this is not always the case. For example, when the



(a) (b)

Figure 3: (a) Bisector surface (in gray) of a convex surface and a plane; and (b) bisector surface (in gray) of a concave surface and a plane.



(a) (b)

Figure 4: Bisector surfaces (in gray) between two freeform surfaces.

| Figure | 1 (a) | 1 (b) | 2 | 3 (a) | 3 (b) | 4 (a) | 4 (b) |
|---|---|---|---|---|---|---|---|
| # Points | 558 | 122 | 132 | 64 | 77 | 318 | 80 |

Table 1: Number of points computed on the zero-set of each example.

two input surfaces intersect each other as shown in Figure 5, both sides of each input surface must be considered in computing the bisector surface. In this example, each point of an input surface corresponds to two different points on the other surface (except those in the intersection curve); thus, each point on

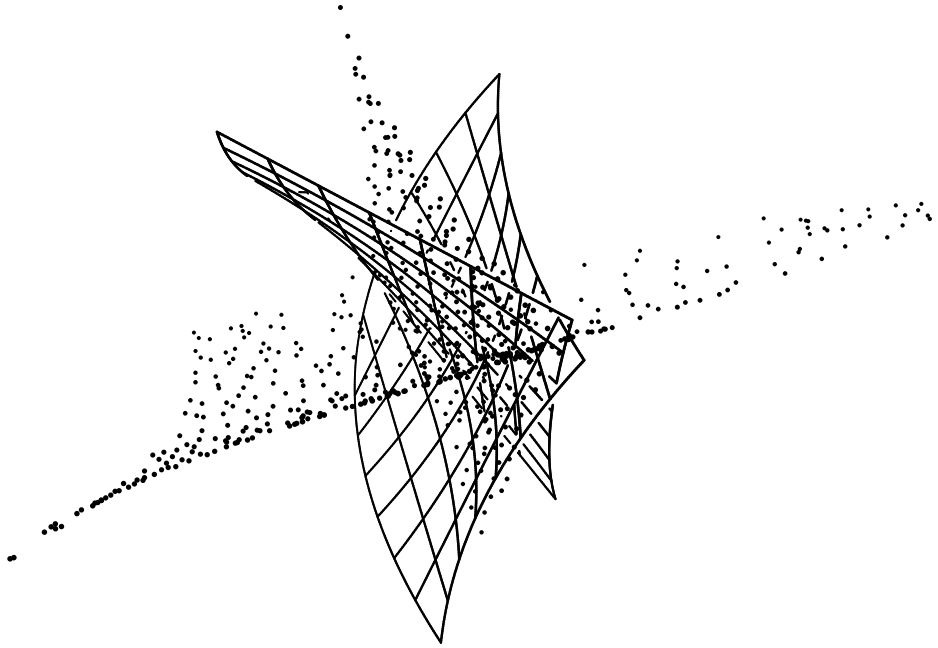


Figure 5: A case of a surface–surface-bisector that produces multiple bisector's sheets and hence are not one to one.

an input surface generates two points on the bisector surface. Note that the bisector surface self-intersects at the intersection curve of two input surfaces. Near the intersection curve, we have difficulty in fitting a surface through discrete points sampled on the bisector surface. We are no longer able to employ the parameterization of an input surface. The fitted surfaces are not single sheeted, but rather two sheets that intersect each other. Nonetheless, as long as the correspondence between the two input surfaces is one-to-one, the simple surface fitting approach proposed in this paper works very well even for the self-intersecting case as shown in Figure 3(b).

## 5 Conclusions

In this paper, we proposed a new computational scheme for the construction of bisector surfaces in $\mathbb{R}^3$, between a rational curve and a rational surface, and also that between two rational surfaces. Although the approach is promising, there are some drawbacks in the current result. In particular, we need to resolve

1. the parameterization of multi-sheeted bisector surfaces, and,

2. the global error measure in the approximation of a bisector surface using surfaces fitted through the discrete sample points.

## Acknowledgments

This research was supported in part by the Fund for Promotion of Research at The Technion, Haifa, Israel, and by Korean Ministry of Science and Technology (MOST) under National Research Laboratory Project.

## References

[1] G. H. Golub and C. F. Van Loan. Matrix Computation. The John Hopkins University Press, Baltimore and London Third Edition, 1996.

[2] D. Dutta and C. Hoffmann. On the skeleton of simple CSG objects. *ASME J. of Mechanical Design*, vol. 115, pp. 87–94, 1993.

[3] G. Elber and M.-S. Kim. The bisector surface of freeform rational space curves. *ACM Trans. on Graphics*, vol. 17, no. 1, pp. 32–49, January 1998.

[4] G. Elber and M.-S. Kim. Bisector curves of planar rational curves. *Computer-Aided Design*, vol. 30, no. 14, pp. 1089–1096, December 1998.

[5] G. Elber and M.-S. Kim. Rational bisectors of CSG primitives. *Proc. 5th ACM/IEEE Symposium on Solid Modeling and Applications*, Ann Arbor, Michigan, pp. 246–257, June 1999.

[6] G. Elber and M.-S. Kim. Computing rational bisectors. *IEEE Computer Graphics and Applications*, to appear. Also Tech. Report TR-CIS-9805, Center for Intelligent Systems, Dept. of Computer Science, The Technion—IIT, Haifa, Israel.

[7] R. Farouki, and J. Johnstone. The bisector of a point and a plane parametric curve. *Computer Aided Geometric Design*, vol. 11, no. 2, pp. 117–151, April 1994.

[8] R. Farouki, and R. Ramamurthy. Specified-precision computation of curve/curve bisectors. *Int. J. of Computational Geometry & Applications*, vol. 8, nos. 5–6, pp. 599–617, October–December 1998.

[9] IRIT 7.0 User's Manual. The Technion—IIT, Haifa, Israel, 1997. Available at `http://www.cs.technion.ac.il/~irit`.

[10] M. Peternell. Geometric properties of bisector surfaces. Submitted to *Graphical Models and Image Processing*, 1999.

[11] R. Ramamurthy. *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.

[12] S. Wolfram. *Mathematica*, 2nd Ed., Addison-Wesley, (1991).