The Minkowski Sum of Two Simple Surfaces Generated by Slope-Monotone Closed Curves

Joon-Kyung Seong and *Myung-Soo Kim*
School of Computer Science and Engineering
Seoul National University
Seoul, Korea

Kokichi Sugihara
Department of Mathematical Informatics
University of Tokyo
Tokyo, Japan

Abstract

We present an algorithm for computing Minkowski sums among surfaces of revolution and surfaces of linear extrusion, generated by slope-monotone closed curves. The special structure of these simple surfaces allows the process of normal matching between two surfaces to be expressed as an explicit equation. Based on this insight, we also present an efficient algorithm for computing the distance between two simple surfaces, even though they may in general be non-convex. Using an experimental implementation, the distance between two surfaces of revolution was computed in less than 0.5 msec on average.

1 Introduction

Computing the Minkowski sum of two objects is closely related to computing the Euclidean distance between them and hence to detecting collisions [3, 8, 20, 21]. Previous work has mainly focused on computing (or utilizing the structure of) the Minkowski sum of (i) two convex free-form objects [2, 8, 16], (ii) two polygonal/polyhedral objects [3, 5, 6, 7, 10, 18, 20, 21, 22], or (iii) two planar free-form objects [1, 12, 16, 17]. It is a very complicated task to compute and represent the more general Minkowski sum of two non-convex three-dimensional free-form objects. Thus it is worthwhile to consider special cases where the Minkowski sum can be computed relatively easily.

In this paper, we present an efficient algorithm for computing the Minkowski sum of two simple surfaces—by which we mean surfaces of revolution and surfaces of linear extrusion generated by slope-monotone closed curves (i.e. curves of continually increasing or continually decreasing slope). Note that a convex region in a plane is bounded by a slope-monotone closed curve; but the latter does not always bound a convex region. In general, a slope-monotone closed curve bounds a set of convex regions; but their union may form a non-convex object (see Figure 1).

The slope-monotonicity of these simple curves allows our algorithm to work much like a Minkowski sum algorithm for convex objects. Moreover, surfaces of revolution and surfaces of linear extrusion are $2\frac{1}{2}$-dimensional, and not full-blown three-dimensional surfaces. An important advantage of these simple surfaces is the simplicity of their Gauss maps, which makes normal matching quite straightforward. Once the normal matching has been done, the Minkowski sum can quite easily be computed as the vector sums of pairs of points, one on each surface, that correspond to the same normals.

The main contribution of this paper is the derivations of explicit formulas for matching the normals of two simple surfaces. These formulas can also be used to generate all points where a surface has a given outward normal direction. Based on this simple method, we have developed an efficient algorithm for computing the distance between two surfaces of revolution. Note again that these surfaces are non-convex in general.

Previous Minkowski sum algorithms deal with non-convex objects by decomposing them into convex pieces [9, 15, 18, 19]. Many real-world objects can be fully or in large part modeled using surfaces of revolution and surfaces of linear extrusion. Consequently, our algorithm has a great deal of potential for real-time collision detection (and hence avoidance) among non-convex three-dimensional objects in everyday use. In an experimental result shown in Figure 14, the distance between the objects at each snapshot was computed in less than 0.5 msec on average, which compares quite favorably with previous results [9, 13, 15] for less general objects.

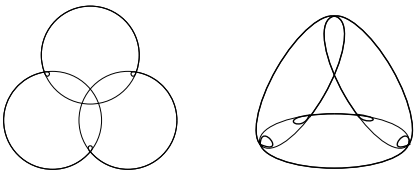
The rest of this paper is organized as follows. In Section 2, we review some preliminary material. Sections 3–5 consider the Minkowski sum of surfaces of revolution and surfaces of linear extrusion. In Section 6, we present an algorithm for computing the distance between two surfaces of revolution. Finally, Section 7 concludes this paper.

2 Preliminaries

We will briefly review slope-monotone closed curves, the surfaces of revolution generated by these curves, and their Minkowski sum. Formal definitions in this area [14, 23] are rather tedious and we take an informal approach.

The slope-monotone closed curve shown in Figure 1(a) represents the union of three circular regions, while that in Figure 1(b) represents the union of three elliptic regions. While tracing along a curve counterclockwise, the region immediately to the left of the curve will be in the interior of the object. The triangular region at the center of Figure 1(b) is not in the interior since this region is always (locally) to the right of the curve.

The slope-monotone closed curve of Figure 1(a) has four local minimum points. For any fixed direction, there are always four locations where the curve has its outward normal in that direction. This number is called the *cycle* of a slope-monotone closed curve. Thus the curve in Figure 1(a) has cycle 4, even though this curve represents a union of three convex regions. The curve in Figure 1(b) has cycle 5.

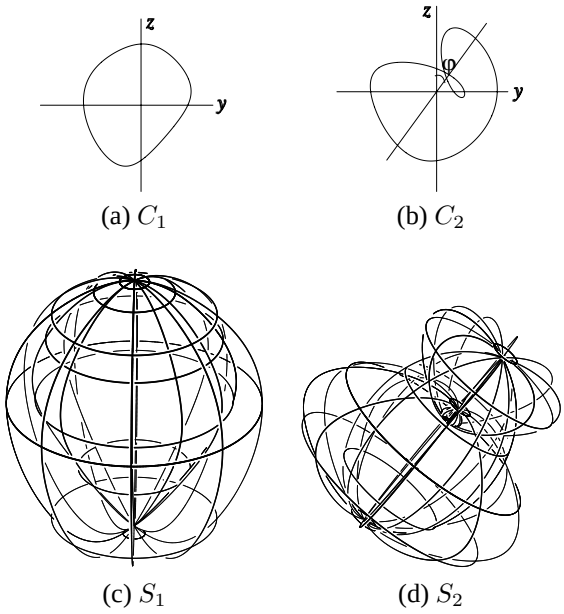


(a) (b)

Figure 1. Slope-monotone closed curves (drawn in bold) representing unions of convex regions.

When we rotate a slope-monotone closed curve about a fixed axis, we obtain a surface of revolution (see Figure 2). (We will assume that the curve is not symmetric about this axis.) Because of the symmetry of the resulting surface shape about its axis of rotation, there are $2k$ points where the surface has its outward normal pointing in a fixed direction N, where k is the cycle of the generating curve. (We will assume that the direction N is not parallel or anti-parallel to the axis of rotation.) In Figure 2(a), the curve has cycle 1; thus the surface S_1 in Figure 2(c) has two points where the surface has its outward normal in a fixed direction $N \neq (0, 0, 1)$. Similarly, the surface S_2 of Figure 2(d) has four points where the surface has its outward normal in a fixed direction.

Given two surfaces of revolution, generated by slope-monotone curves of cycle k_i, $i = 1, 2$, each surface has $2k_i$ points where its outward normal is in a given direction N. By matching these points and taking their vector sums, we generate $4k_1k_2$ points; as we change the normal direction N smoothly they in turn generate surface patches. These comprise the Minkowski sum surface, which bounds the volumetric Minkowski sum $V_1 \oplus V_2$ of two solids V_1 and V_2, where each V_i is the volume bounded by S_i, and $V_1 \oplus V_2 = \{\mathbf{v}_1 + \mathbf{v}_2 \mid \mathbf{v}_i \in V_i\}$, $i = 1, 2$. (See Kim and Sugihara [14] for more details.)



(a) C_1 (b) C_2

(c) S_1 (d) S_2

Figure 2. Two slope-monotone closed curves and their surfaces of revolution.

Depending on the way in which solid objects are being represented (e.g. as boundary models), a global trimming procedure may be needed to complete the construction, by eliminating the redundant parts of the Minkowski sum surface that belong to the interior of the Minkowski sum [17]. Although trimming is a non-trivial task, we ignore the details in this paper. Depending on the application, it is often sufficient to do trimming locally, indirectly, or only approximately. In the following sections, the Minkowski sum of two surfaces means the whole of the Minkowski sum surface generated by the vector sums of surface points corresponding to the same surface normals.

3 Two Surfaces of Revolution

We now present an algorithm for computing the Minkowski sum of two surfaces of revolution generated by slope-monotone closed curves. This result is an extension of Kim and Sugihara [14], where the Minkowski sum was computed for two axis-parallel surfaces of revolution.

3.1 Normal Vectors

Two slope-monotone closed curves C_1 and C_2 are defined in the yz-plane as follows:

$$C_1(s) = (0, y_1(s), z_1(s)),\ C_2(t) = (0, y_2(t), z_2(t)).$$

An axis l is contained in the yz-plane and makes an angle φ with the z-axis. Let S_1 denote the surface generated by rotating the curve C_1 about the z-axis; and let S_2 denote another surface generated by rotating C_2 about the axis l:

$$\begin{aligned} S_1(\theta, s) &= (-y_1(s)\sin\theta, y_1(s)\cos\theta, z_1(s)), \\ S_2(\psi, t) &= (-\widehat{y}_2(t)\sin\psi, \widehat{y}_2(t)\cos\psi\cos\varphi + \widehat{z}_2(t)\sin\varphi, \\ &\quad -\widehat{y}_2(t)\cos\psi\sin\varphi + \widehat{z}_2(t)\cos\varphi), \end{aligned}$$

where

$$\begin{aligned} \widehat{y}_2(t) &= y_2(t)\cos\varphi - z_2(t)\sin\varphi, \\ \widehat{z}_2(t) &= y_2(t)\sin\varphi + z_2(t)\cos\varphi, \end{aligned}$$

for all θ, ψ, s, and t (see Figure 2). Note that the parameterization of $S_2(\psi, t)$ can be obtained in three steps: (i) rotate $C_2(t)$ through an angle φ about the x-axis and call the resulting curve $\widehat{C}_2(t) = (0, \widehat{y}_2(t), \widehat{z}_2(t))$, (ii) rotate $\widehat{C}_2(t)$ through an angle ψ about the z-axis, and (iii) rotate the resulting surface of revolution through an angle $-\varphi$ about the x-axis.

From the partial derivatives of $S_1(\theta, s)$,

$$\begin{aligned} \frac{\partial S_1}{\partial \theta} &= (-y_1(s)\cos\theta, -y_1(s)\sin\theta, 0), \\ \frac{\partial S_1}{\partial s} &= (-y_1'(s)\sin\theta, y_1'(s)\cos\theta, z_1'(s)), \end{aligned}$$

we can compute the normal vector of $S_1(\theta, s)$ as

$$\frac{\partial S_1}{\partial \theta} \times \frac{\partial S_1}{\partial s} = y_1(s)(-z_1'(s)\sin\theta, z_1'(s)\cos\theta, -y_1'(s)).$$

Similarly, from the partial derivatives

$$\begin{aligned} \frac{\partial S_2}{\partial \psi} &= (-\widehat{y}_2(t)\cos\psi, -\widehat{y}_2(t)\sin\psi\cos\varphi, \\ &\quad \widehat{y}_2(t)\sin\psi\sin\varphi), \\ \frac{\partial S_2}{\partial t} &= (-\widehat{y}_2'(t)\sin\psi, \widehat{y}_2'(t)\cos\psi\cos\varphi + \widehat{z}_2'(t)\sin\varphi, \\ &\quad -\widehat{y}_2'(t)\cos\psi\sin\varphi + \widehat{z}_2'(t)\cos\varphi), \end{aligned}$$

we can compute the normal vector of $S_2(\psi, t)$ as

$$\begin{aligned} &\frac{\partial S_2}{\partial \psi} \times \frac{\partial S_2}{\partial t} = \\ &\widehat{y}_2(t)(-z_2'(t)\sin\psi, z_2'(t)\cos\psi\cos\varphi - \widehat{y}_2'(t)\sin\varphi, \\ &\quad -\widehat{y}_2'(t)\cos\varphi - \widehat{z}_2'(t)\cos\psi\sin\varphi). \end{aligned} \tag{1}$$

Note that, when $C_1(s)$ passes through the z-axis (and thus $y_1(s)$ changes its sign), the normal vector changes its direction. Moreover, when $y_1(s) = 0$, the normal vector vanishes. Thus we define an outward normal vector $\overline{N}_1(\theta, s)$ by deleting the term $y_1(s)$, to obtain:

$$\overline{N}_1(\theta, s) = (-z_1'(s)\sin\theta, z_1'(s)\cos\theta, -y_1'(s)).$$

Let $N_1(\theta, s)$ denote the unit outward normal vector of $S_1(\theta, s)$. It can be formulated as:

$$N_1(\theta, s) = \frac{(-z_1'(s)\sin\theta, z_1'(s)\cos\theta, -y_1'(s))}{\sqrt{y_1'(s)^2 + z_1'(s)^2}}.$$

We can repeat this procedure for the other surface $S_2(\psi, t)$. In this case the axis l plays a similar role to that of the z-axis in the discussion above, and $N_2(\psi, t)$ denotes the unit outward normal vector of $S_2(\psi, t)$.

3.2 Matching Normal Vectors

To move nearer to computing the Minkowski sum, we now need to find all pairs of matching normal vectors of S_1 and S_2. That is, given θ and s, we need to compute the parameters ψ and t that satisfy the following relation:

$$N_1(\theta, s) = N_2(\psi, t).$$

This procedure involves computation of the Gauss maps of S_1 and S_2, which is relatively easy because the surfaces we are dealing with are generated by slope-monotone closed curves. For each s, the normal vectors $N_1(\theta, s)$, $(0 \le \theta \le 2\pi)$, generate a circle on the Gauss sphere. Similarly, for each t, the normal vectors $N_2(\psi, t)$, $(0 \le \psi \le 2\pi)$, generate another circle on the Gauss sphere (see Figure 3).

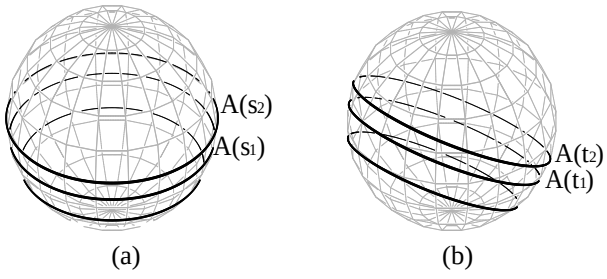


Figure 3. Circles on the Gauss sphere: (a) circles generated by $N_1(\theta, s)$, and (b) circles generated by $N_2(\psi, t)$.

Given θ and s, we now consider how to compute the parameters ψ and t that satisfy $N_1(\theta, s) = N_2(\psi, t)$. Note

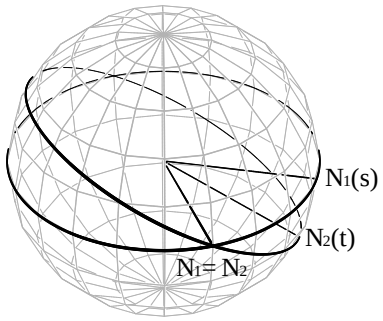


Figure 4. Matching normal vectors.

that $N_2(0,t)$ is located on the yz-plane, and $N_2(\psi,t)$ is obtained by rotating $N_2(0,t)$ about the axis l by angle ψ. Thus we have

$$\begin{aligned} N_2(0,t) &= R_l(-\psi)N_2(\psi,t) \\ &= R_l(-\psi)N_1(\theta,s), \end{aligned} \tag{2}$$

where $R_l(\psi)$ represents the rotation about axis l by angle ψ.

The normal vector $N_2(0,t)$ is computed using Equation (2):

$$\begin{aligned} & N_2(0,t) = R_l(-\psi)N_1(\theta,s) \\ = \; & \frac{1}{\sqrt{y_1'(s)^2 + z_1'(s)^2}} \\ & \begin{pmatrix} -z_1'(s)(\sin\theta\cos\psi - \cos\theta\cos\varphi\sin\psi) + \\ y_1'(s)\sin\varphi\sin\psi \\ \\ z_1'(s)(\sin\theta\sin\psi\cos\varphi + \\ \cos\theta\cos^2\varphi\cos\psi + \cos\theta\sin^2\varphi) + \\ y_1'(s)(\sin\varphi\cos\varphi\cos\psi - \sin\varphi\cos\varphi) \\ \\ -z_1'(s)\sin\theta\sin\psi\sin\varphi - \\ z_1'(s)\cos\theta(\sin\varphi\cos\varphi\cos\psi - \cos\varphi\sin\varphi) - \\ y_1'(s)(\sin^2\varphi\cos\psi + \cos^2\varphi) \end{pmatrix}. \end{aligned} \tag{3}$$

Since $N_2(0,t)$ is contained in the yz-plane, its x-component is equal to zero. Consequently, we have the following equation:

$$\begin{aligned} -z_1'(s)(\sin\theta\cos\psi - \cos\theta\cos\varphi\sin\psi) \\ + y_1'(s)\sin\varphi\sin\psi = 0, \end{aligned}$$

which produces

$$\tan\psi = \frac{z_1'(s)\sin\theta}{y_1'(s)\sin\varphi + z_1'(s)\cos\theta\cos\varphi}. \tag{4}$$

When $\cos\psi \approx 0$, we can derive $\cot\psi$ instead of $\tan\psi$. Given θ and s, the angle $\psi = \psi(\theta,s)$ is computed as a function of θ and s from Equation (4); and we are able to find the other parameter $t = t(\theta,s)$ and the curve point $C_2(t)$, using Equation (3). There are two solutions to Equation (4) in the range $[0,2\pi]$: ψ_a and $\psi_b = \psi_a + \pi$. Moreover, for a fixed $\psi = \psi_a$ or $\psi = \psi_b$, there are k_2 different values of t that produce the same unit normal vector $N_2(\psi,t)$, where k_2 is the cycle of $C_2(t)$. This means that there are $2k_2$ different pairs (ψ,t) that produce the same $N_2(\psi,t)$, where $\psi = \psi_a$ or $\psi = \psi_b$. A similar argument applies to $N_1(\theta,s)$. Consequently, we have $4k_1k_2$ pairs of matching normal vectors for each normal direction, except those parallel or anti-parallel to the rotational axes of S_1 and S_2.

3.3 Computing the Minkowski Sum

Let $N_2(0,t_a(\theta,s))$ be the normal vector computed in Equation (3) using the angle $\psi_a(\theta,s)$; and let $N_2(0,t_b(\theta,s))$ be the normal vector computed from the angle $\psi_b(\theta,s) = \psi_a(\theta,s) + \pi$. Then, from the following normal matching conditions

$$\begin{aligned} N_1(\theta,s) &= N_2(\psi_a(\theta,s), t_a(\theta,s)), \\ N_1(\theta,s) &= N_2(\psi_b(\theta,s), t_b(\theta,s)), \end{aligned}$$

we can define two partial Minkowski sums as follows:

$$\begin{aligned} (S_1 \oplus S_2)_a(\theta,s) &= S_1(\theta,s) + S_2(\psi_a(\theta,s), t_a(\theta,s)), \\ (S_1 \oplus S_2)_b(\theta,s) &= S_1(\theta,s) + S_2(\psi_b(\theta,s), t_b(\theta,s)), \end{aligned}$$

for all θ and s. The Minkowski sum of $S_1(\theta,s)$ and $S_2(\psi,t)$ is then defined as the union of these two surfaces,

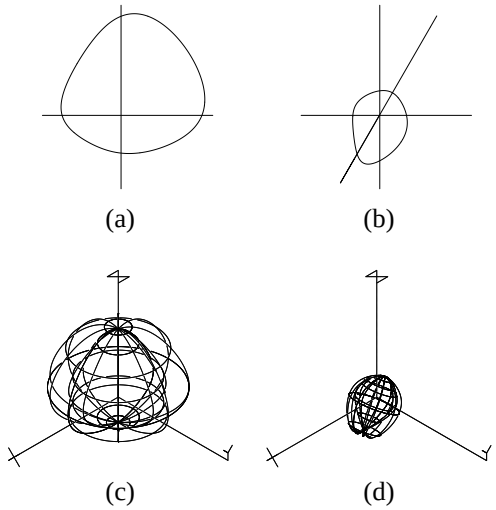
$$(S_1 \oplus S_2)(\theta,s) = (S_1 \oplus S_2)_a(\theta,s) \cup (S_1 \oplus S_2)_b(\theta,s),$$

for all θ and s.

Let $R(S_i)$, $i = 1,2$, denote the three-dimensional volume which is enclosed by the surface S_i. When the axes of rotation are non-parallel, the three-dimensional volumetric Minkowski sum is given as

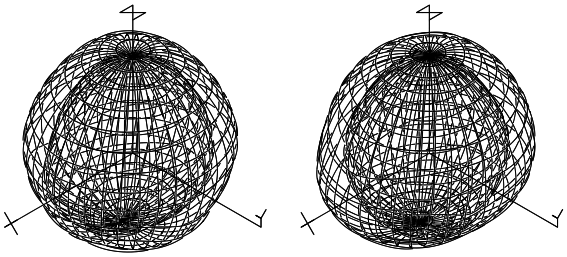
$$\begin{aligned} & R(S_1) \oplus R(S_2) \\ = \; & \{\, \mathbf{p}_1 + \mathbf{p}_2 \mid \mathbf{p}_1 \in R(S_1),\ \mathbf{p}_2 \in R(S_2)\} \\ = \; & R((S_1 \oplus S_2)_a) \cup R((S_1 \oplus S_2)_b). \end{aligned}$$

(If the two axes are parallel, there will be some extra volumes to consider; see Kim and Sugihara [14] for more details and some examples.) Figure 5 shows two surfaces of revolution generated by slope-monotone closed curves. Figure 6 shows their partial Minkowski sums. Figure 7 shows two tori and their partial Minkowski sums.



(a) (b) (c) (d)

Figure 5. Two slope-monotone closed curves and their surfaces of revolution.

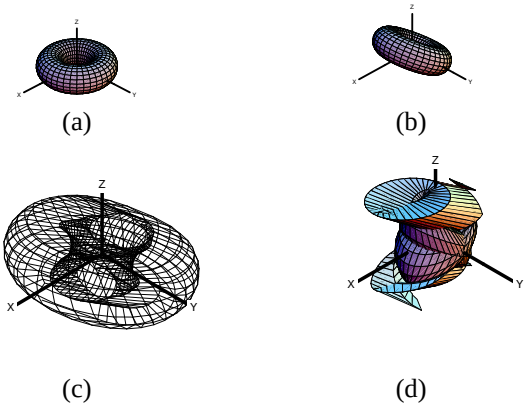


(a) (b)

Figure 6. The partial Minkowski sums of two surfaces of revolution.

4 Two Surfaces of Linear Extrusion

We now move on to consider the Minkowski sum of two surfaces of linear extrusion, again generated by slope-monotone closed curves. The outward unit normals of a surface of linear extrusion are orthogonal to the direction of extrusion and thus they form a great circle on the Gauss sphere. Consequently, the Minkowski sum is relatively easy to construct: again, normal matching is done by intersecting the two great circles on the Gauss sphere.



(a) (b) (c) (d)

Figure 7. Two tori and their partial Minkowski sums.

4.1 Normal Vectors

Two slope-monotone closed curves are now defined in the xy-plane as:

$$C_1(s) = (x_1(s), y_1(s), 0),\ C_2(t) = (x_2(t), y_2(t), 0).$$

A fixed direction $\mathbf{l}_1 = (a_1, 0, c_1)$ is given on the xz-plane and another direction $\mathbf{l}_2 = (a_2, b_2, c_2)$ is chosen arbitrarily. We define two surfaces of linear extrusion as follows:

$$\begin{aligned} S_1(u,s) &= C_1(s) + u\mathbf{l}_1 \\ &= (x_1(s) + ua_1, y_1(s), uc_1), \\ S_2(v,t) &= C_2(t) + v\mathbf{l}_2 \\ &= (x_2(t) + va_2, y_2(t) + vb_2, vc_2). \end{aligned}$$

From the partial derivatives of S_1 and S_2, which are

$$\frac{\partial S_1}{\partial s} = (x_1'(s), y_1'(s), 0), \quad \tfrac{\partial S_1}{\partial u} = (a_1, 0, c_1),$$
$$\frac{\partial S_2}{\partial t} = (x_2'(t), y_2'(t), 0), \quad \tfrac{\partial S_2}{\partial v} = (a_2, b_2, c_2),$$

we can compute the unit outward normal vectors N_1 and N_2, as follows:

$$\begin{aligned} N_1(u,s) &= \overline{N}_1(u,s)/\|\overline{N}_1(u,s)\|, \\ N_2(v,t) &= \overline{N}_2(v,t)/\|\overline{N}_2(v,t)\|, \end{aligned}$$

where

$$\begin{aligned} \overline{N}_1(u,s) &= (c_1 y_1'(s), -c_1 x_1'(s), -a_1 y_1'(s)), \\ \overline{N}_2(v,t) &= (c_2 y_2'(t), -c_2 x_2'(t), b_2 x_2'(t) - a_2 y_2'(t)). \end{aligned}$$

Figure 8 shows the images of $N_1(u,s)$ and $N_2(v,t)$ which generate great circles on the Gauss sphere. Note that each circle is also contained in the plane orthogonal to the extrusion direction $\mathbf{l}_i$.

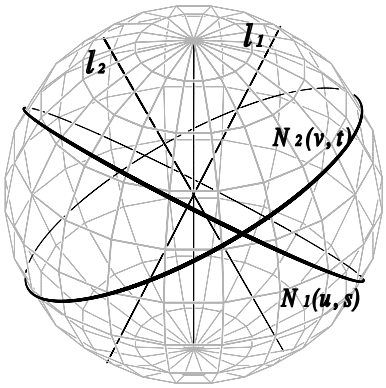


Figure 8. Gauss images of the surfaces of linear extrusion.

4.2 Matching Normal Vectors

In Figure 8, two great circles intersect at two antipodal points. When the extrusion directions are not parallel, the two surfaces of linear extrusion have matching normal vectors only at the directions $\pm \mathbf{l}_1 \times \mathbf{l}_2$. On the other hand, when the directions are parallel, the problem essentially reduces to a planar case, which can be solved using the technique of Sugihara et al. [23].

4.3 Computing the Minkowski Sum

Let $N_\pm$ denote two unit normal directions defined as

$$N_\pm = \pm \frac{\mathbf{l}_1 \times \mathbf{l}_2}{\|\mathbf{l}_1 \times \mathbf{l}_2\|}.$$

For the direction N_+, there are $k_1 k_2$ pairs of curve points $(C_1(s_{i,+}), C_2(t_{j,+}))$ where the surfaces S_1 and S_2 have N_+ as the unit outward normal vector, for $i = 1, \ldots, k_1$, and $j = 1, \ldots, k_2$. (Recall that k_l is the cycle of a slope-monotone closed curve C_l.) Analogous arguments apply to the direction N_-.

Let $\overline{N}_\pm$ denote the projection of $N_\pm$ onto the xy-plane. Then the curve $C_1(s)$ has $\overline{N}_+$ as its outward normal vector at each $s = s_{i,+}$; and similarly the curve $C_1(s)$ has $\overline{N}_-$ as its outward normal vector at each $s = s_{i,-}$. Analogous arguments apply to the other curve $C_2(t)$ and the parameters $t_{j,+}$ and $t_{j,-}$.

The Minkowski sum of S_1 and S_2 consists of $2k_1k_2$ planes, each of which is orthogonal to $\mathbf{l}_1 \times \mathbf{l}_2$ and contains one of the points: $C_1(s_{i,+}) + C_2(t_{j,+})$ or $C_1(s_{i,-}) + C_2(t_{j,-})$, for $i = 1, \ldots, k_1$, and $j = 1, \ldots, k_2$. Here we consider surfaces of linear extrusion, which are infinitely extended. The case of truncated surfaces is more involved; to save space, we omit this more general case.

5 A Surface of Revolution and a Surface of Linear Extrusion

In this section, we consider the Minkowski sum of a surface of revolution and a surface of linear extrusion, where the surfaces are generated by slope-monotone closed curves.

5.1 Normal Vectors

Two generating curves are defined as follows:

$$C_1(s) = (0, y_1(s), z_1(s)),\ C_2(t) = (x_2(t), y_2(t), 0).$$

Then the surface of revolution S_1 may be parameterized as

$$S_1(\theta, s) = (-y_1(s)\sin\theta,\ y_1(s)\cos\theta,\ z_1(s)),$$

and the surface of linear extrusion S_2 is given as

$$S_2(u,t) = C_2(t) + u\mathbf{l} = (x_2(t) + ua, y_2(t) + ub, uc),$$

where $\mathbf{l} = (a, b, c)$ is a direction vector. Their unit outward normal vectors are

$$\begin{aligned} N_1(\theta, s) &= \frac{(-z_1'(s)\sin\theta, z_1'(s)\cos\theta, -y_1'(s))}{\sqrt{y_1'(s)^2 + z_1'(s)^2}}, \\ N_2(u,t) &= \overline{N}_2(u,t)/\|\overline{N}_2(u,t)\|, \end{aligned}$$

where

$$\overline{N}_2(u,t) = (cy_2'(t),\ -cx_2'(t),\ bx_2'(t) - ay_2'(t)).$$

5.2 Matching Normal Vectors

Given u and t, we consider how to compute the parameters θ and s so that the two unit normal vectors $N_1(\theta, s)$ and $N_2(u,t)$ are matched. We can deduce the normal vector $N_1(0, s)$ by rotating $N_1(\theta, s)$ through an angle $-\theta$ about the z-axis. Thus, we have

$$\begin{aligned} N_1(0,s) &= R_z(-\theta) N_1(\theta, s) \\ &= R_z(-\theta) N_2(u,t) \\ &= (cy_2'(t)\cos\theta - cx_2'(t)\sin\theta,\ -cy_2'(t)\sin\theta \\ &\quad -cx_2'(t)\cos\theta, bx_2'(t) - ay_2'(t)). \end{aligned} \tag{5}$$

Figure 9 shows the Gauss maps of these two surfaces overlayed on the same Gauss sphere. Since the x-component of $N_1(0, s)$ is equal to zero, we deduce that

$$\tan\theta = y_2'(t)/x_2'(t). \tag{6}$$

When $\cos\theta \approx 0$, we can use $\cot\theta$ instead of $\tan\theta$. Given u and t, the angle $\theta = \theta(u,t)$ is computed as a function of u and t from Equation (6); and we are able to find the corresponding value of $s = s(u,t)$ from Equation (5). Note that there are $2k_1k_2$ pairs of matching normal vectors, excluding the directions parallel or anti-parallel to the axis of rotation or the direction of extrusion.

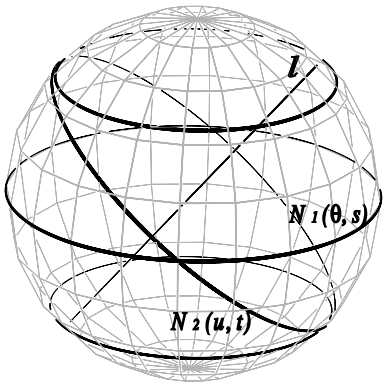


Figure 9. Two Gauss maps overlayed.

5.3 Computing the Minkowski Sum

The Minkowski sum of $S_1(\theta, s)$ and $S_2(u, t)$ can be constructed as a union of partial Minkowski sums. From the relation

$$N_1(\theta_i(u,t), s_i(u,t)) = N_2(u,t), \quad i = 1, ..., 2k_1,$$

we can define partial Minkowski sums as follows:

$$(S_1 \oplus S_2)_i(u,t) = S_1(\theta_i(u,t), s_i(u,t)) + S_2(u,t),$$

for all u and t, and $i = 1, ..., 2k_1$.

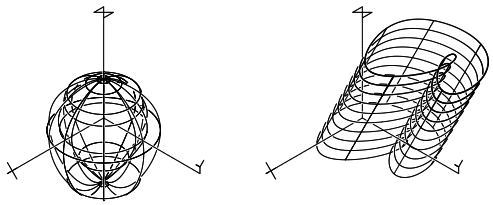
Figure 10 shows a surface of revolution and a surface of linear extrusion. Note that the surface of revolution is generated by a slope-monotone curve of cycle 1. Thus two partial Minkowski sums are generated, as shown in Figure 11.

6 Computing Distance

The Minkowski sum can be used in computing the distance between two surfaces. Given two surfaces S_1 and S_2, we may assume that the distance between S_1 and S_2 is realized at two surface (interior) points $\mathbf{p}_1 \in S_1$ and $\mathbf{p}_2 \in S_2$. Then $\mathbf{p}_1 - \mathbf{p}_2$ is the closest point from the origin to the Minkowski sum $S_1 \oplus (-S_2)$. Consider a ball of radius $\|\mathbf{p}_1 - \mathbf{p}_2\|$, with its center at the origin; this ball will touch the Minkowski sum $S_1 \oplus (-S_2)$ tangentially at $\mathbf{p}_1 - \mathbf{p}_2$.

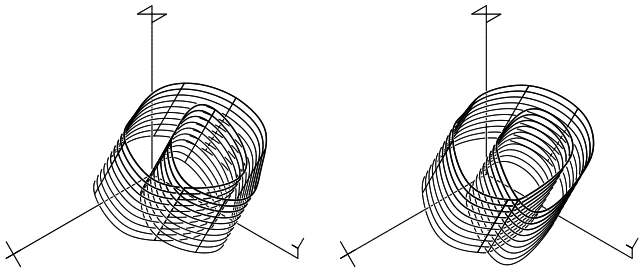
Let $N_i(\mathbf{p}_i)$ denote the unit outward normal vector of S_i at $\mathbf{p}_i$, $i = 1, 2$. Then $N_1(\mathbf{p}_1)$ is the same as the unit outward normal vector of $S_1 \oplus (-S_2)$ at $\mathbf{p}_1 - \mathbf{p}_2$, whereas $N_2(\mathbf{p}_2)$ is anti-parallel to $N_1(\mathbf{p}_1)$. Moreover, the direction vector $\mathbf{p}_1 - \mathbf{p}_2$ is parallel to $N_2(\mathbf{p}_2)$ and anti-parallel to $N_1(\mathbf{p}_1)$.

Normal matching between S_1 and $-S_2$ is a first step in computing the distance $\|\mathbf{p}_1 - \mathbf{p}_2\|$ between the two surfaces S_1 and S_2. In the next step, we need to check whether the direction vector $\mathbf{p}_1 - \mathbf{p}_2$ is parallel to $N_2(\mathbf{p}_2)$; if not, some other pair $(\mathbf{p}_1, \mathbf{p}_2)$ of points should be tested. Once all these conditions have been met, we have finally to make sure that the two points $\mathbf{p}_1$ and $\mathbf{p}_2$ are really the closest points (i.e. that there are no other closer points).



(a) $S_1(\theta, s)$ (b) $S_2(u, t)$

Figure 10. A surface of revolution and a surface of linear extrusion.



(a) $(S_1 \oplus S_2)_1(u, t)$ (b) $(S_1 \oplus S_2)_2(u, t)$

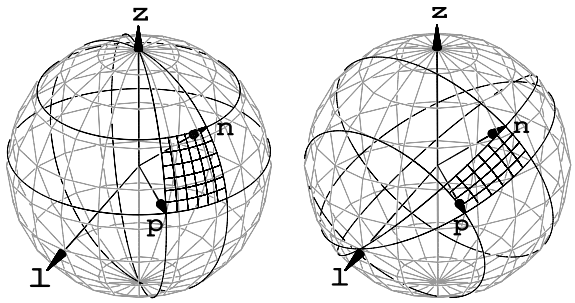
Figure 11. The Minkowski sum of a surface of revolution and a surface of linear extrusion.

We could perform normal matching for a coverage of points across the whole of each surface; but this is unnecessary, since we are only trying to find the two closest points. Starting with a good initial guess, the search can be carried out quite efficiently using an iterative procedure. And the explicit normal matching formula further simplifies this procedure.

We will now describe a heuristic algorithm for finding the closest points between two surfaces of revolution. First of all, we guess a unit normal direction $\mathbf{n}$ which, we hope, is nearly parallel to $\mathbf{p}_1 - \mathbf{p}_2$. There are $2k_1$ points on S_1 where the surface has $\mathbf{n}$ as its outward normal direction. Similarly, there are $2k_2$ points on S_2 where the surface has $-\mathbf{n}$ as its outward normal direction. Among $4k_1k_2$ pairs of these points, we select the pair $(\mathbf{p}_1, \mathbf{p}_2)$ with the smallest distance $\|\mathbf{p}_1 - \mathbf{p}_2\|$ between them.

If the direction of $\mathbf{p}_1 - \mathbf{p}_2$ is indeed parallel to $\mathbf{n}$, we

are done. Otherwise, we have to modify the normal direction $\mathbf{n}$ so as to improve the approximation to the closest points $\mathbf{p}_1$ and $\mathbf{p}_2$. The slope-monotonicity of our generating curve greatly simplifies this search procedure. Let $p = (\mathbf{p}_1 - \mathbf{p}_2)/\|\mathbf{p}_1 - \mathbf{p}_2\|$, and consider the Gauss maps of the two surfaces S_1 and $-S_2$. Figure 12(a) shows a region determined by two points $\mathbf{n}$ and $\mathbf{p}$ on the Gauss sphere of the surface S_1; Figure 12(b) shows a similar region for the other surface $-S_2$. The final solution, where $\mathbf{n}$ and $\mathbf{p}$ converge, should be located in both regions of the Gauss map shown in Figures 12(a)–12(b). The intersection of these two regions is shown in Figure 13, where the intersected region is bounded by four great circles. In the next iteration, we consider the great circular arc that connects $\mathbf{n}$ and $\mathbf{p}$ on the sphere, and take the location that subdivides this arc in the ratio of $1 : 2$. (We have determined experimentally that this ratio produces a faster convergence than the 'obvious' ratio of $1 : 1$, which sometimes causes the search to oscillate.)



(a) (b)

Figure 12. (a) A region in the Gauss map of S_1, and (b) a region in the Gauss map of S_2.

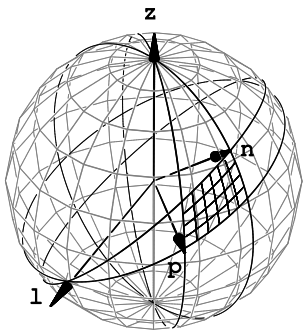


Figure 13. An intersected region bounded by four circular arcs.

Using this simple technique to reduce the size of the search region at each iteration, we can find a candidate pair of closest points on each surface. To make sure that this distance is really the global minimum distance between the two surfaces, we can check $2k_1$ points on S_1, at which the surface has $\mathbf{n}$ as its outward normal direction. If there is any point among these $2k_1$ that is closer to the other surface S_2 than the candidate $\mathbf{p}_1$, we replace $\mathbf{p}_1$ by the point closest to S_2. We then repeat the same procedure for the other surface S_2 using $-\mathbf{n}$ and $\mathbf{p}_2$. If either $\mathbf{p}_1$ or $\mathbf{p}_2$ is updated at this stage, we need to repeat the whole procedure. Otherwise, we have computed the distance between S_1 and S_2.

We have implemented this iterative search algorithm. Figure 14 shows an example where a surface of revolution moves around a torus while smoothly changing its orientation. The figure shows four 'snapshots' of the motion, at each of which the distance between the surfaces was computed. This took less than 0.5 msec on average (on a 500 MHz Linux machine) using a termination condition of $\langle \mathbf{n}, \mathbf{p} \rangle > 0.99999$. This performance demonstrates the potential of this approach in real-time collision detection and avoidance.

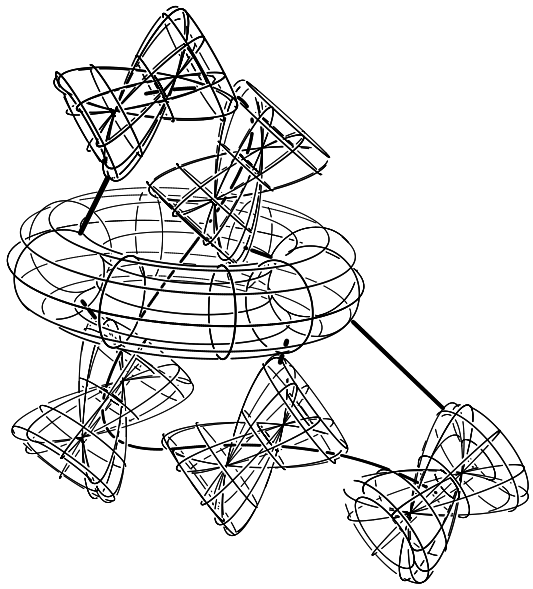


Figure 14. An example distance computation. The bold curved line shows the trajectory of the center of the moving surface, while bold line segments show the pairs of points realizing the shortest distances.

7 Conclusions

We have presented an efficient algorithm for constructing the Minkowski sum of two simple surfaces generated by slope-monotone closed curves. Based on the simple structure of these surfaces, we have formulated explicit equations for matching their surface normals. To demonstrate the effectiveness of this approach, we have implemented an algorithm for computing the distance between two such surfaces of revolution. Experimental results indicate that our approach compares favorably with previous methods. In future work, we would like to extend the geometrical coverage of our method.

Acknowledgements

The authors would like to thank the anonymous reviewers for their useful comments. All the algorithms and figures presented in this paper were implemented and generated using the IRIT solid modeling system [4] developed at the Technion, Israel. This research was supported in part by the Korean Ministry of Information and Communication (MIC) under the program of IT Research Center for CGVR, and in part by the Korean Ministry of Science and Technology (MOST) under the National Research Lab Project.

References

[1] C. Bajaj and M.-S. Kim. Generation of configuration space obstacles : The case of moving algebraic curves. *Algorithmica*, 4(2):157–172, 1989.

[2] C. Bajaj and M.-S. Kim. Generation of configuration space obstacles : The case of moving algebraic surfaces. *The Int'l J. of Robotics Research*, 9(1):92–112, 1990.

[3] S. Cameron. A comparison of two fast algorithm for computing the distance between convex polyhedra. *IEEE Trans. on Robotics and Automation*, 13(6):915–920, 1997.

[4] G. Elber. *IRIT 7.0 User's Manual.* The Technion-IIT, Haifa, Israel, 1997. Available at http://www.cs.technion.ac.il/ irit.

[5] P. Ghosh. A mathematical model for shape description using Minkowski operators. *Computer Vision, Graphics, and Image Processing*, 44(3):239–269, 1988.

[6] P. Ghosh. An algebra of polygons through the notion of negative shapes. *Computer Vision, Graphics, and Image Processing: Image Understanding*, 54(1):119–144, 1991.

[7] P. Ghosh. A unified computational framework for Minkowski operations. *Computers and Graphics*, 17(4):357–378, 1993.

[8] E. Gilbert and C. Foo. Computing the distance between general convex objects in three-dimensional space. *IEEE Trans. on Robotics and Automation*, 6(1):53–61, 1990.

[9] S. Gottschalk, M. Lin and D. Manocha. OBB-Tree: A hierarchical structure for rapid interference detection, *Proc. of ACM SIGGRAPH'96*, 171–180, 1996.

[10] L. Guibas, L. Ramshaw, and J. Stolfi. A kinetic framework for computer geometry. *Proc. of 24th Annual Symp. on Foundations of Computer Science*, 100–111, 1983.

[11] P. Jiménez, F. Thomas, and C. Torras. 3D collision detection: a survey. *Computers & Graphics*, 25:269–285, 2001.

[12] A. Kaul and R. Farouki. Computing Minkowski sums of plane curves. *Int' J. of Computational Geometry & Applications*, 5(4):413–432, 1995.

[13] K. Kawachi and H. Suzuki. Distance computation between non-convex polyhedra at short range based on discrete Voronoi regions. *Proc. of Geometric Modeling and Processing*, Hong Kong, 123–128, 2000.

[14] M.-S. Kim and K. Sugihara. The Minkowski sum of two axis-parallel surfaces of revolution generated by slope-monotone closed curves. *IEICE Trans. on Information and Systems*, E84-D(11):1540–1547, 2001.

[15] J.T. Klosowski, M. Held, J.S.B. Mitchell, H. Sowizral, and K. Zikan. Efficient collision detection using bounding volume hierarchies of k-DOPs. *IEEE Trans. on Visualization and Computer Graphics*, 4(1):21–36, 1998.

[16] M. Kohler and M. Spreng. Fast computation of the C-space of convex 2D algebraic objects. *The Int' J. of Robotics Research*, 14(6):590–608, 1995.

[17] I.-K. Lee, M.-S. Kim, and G. Elber. Polynomial/rational approximation of Minkowski sum boundary curves. *Graphical Models and Image Processing*, 60(2):136–165, 1998.

[18] M.C. Lin and J.F. Canny. A fast algorithm for incremental distance calculation. *Proc. of IEEE Int'l Conference on Robotics and Automation*, Sacramento, California, 1008–1014, 1991.

[19] M.C. Lin and S. Gottschalk. Collision detection between geometric models: a survey. *Mathematics of Surfaces VIII*, R. Cripps, editor, Information Geometers, 37–56, 1998.

[20] T. Lozano-Pérez and M. A. Wesley. An algorithm for planning collision free paths among polyhedral obstacles. *Comm. of the ACM*, 22(10):560–570, 1979.

[21] T. Lozano-Pérez. Spatial planning : A configuration space approach. *IEEE Trans. on Computers*, 32(2):108–120, 1983.

[22] J. Rossignac and A. Kaul. AGRELs and BIPs: Metamorphosis as a Bezier curve in the space of polyhedra. *Computer Graphics Forum*, 13(3):C179–C184, 1994.

[23] K. Sugihara, T. Imai, and T. Hataguchi. An algebra for slope-monotone closed curves. *Int'l J. of Shape Modeling*, 3(3–4):167–183, 1997.