Collision Prediction for Polyhedra under Screw Motions

Byungmoon Kim
bmkim@cc.gatech.edu

Jarek Rossignac
jarek@cc.gatech.edu

GVU Center and College of Computing
Georgia Institute of Technology

ABSTRACT

The prediction of collisions amongst N rigid objects may be reduced to a series of computations of the time to first contact for all pairs of objects. Simple enclosing bounds and hierarchical partitions of the space-time domain are often used to avoid testing object-pairs that clearly will not collide. When the remaining pairs involve only polyhedra under straight-line translation, the exact computation of the collision time and of the contacts requires only solving for intersections between linear geometries. When a pair is subject to a more general relative motion, such a direct collision prediction calculation may be intractable. The popular brute force collision detection strategy of executing the motion for a series of small time steps and of checking for static interferences after each step is often computationally prohibitive. We propose instead a less expensive collision prediction strategy, where we approximate the relative motion between pairs of objects by a sequence of screw motion segments, each defined by the relative position and orientation of the two objects at the beginning and at the end of the segment. We reduce the computation of the exact collision time and of the corresponding face/vertex and edge/edge collision points to the numeric extraction of the roots of simple univariate analytic functions. Furthermore, we propose a series of simple rejection tests, which exploit the particularity of the screw motion to immediately decide that some objects do not collide or to speed-up the prediction of collisions by about 30%, avoiding on average 3/4 of the root-finding queries even when the object actually collide.

Categories and Subject Descriptors

J.6 [**Computer-Aided Engineering**]: Computer-Aided Design

General Terms

Theory

Keywords

Collision Detection, Screw Motion, Polyhedra

SM'03, June 16–20, 2003, Seattle, Washington, USA.

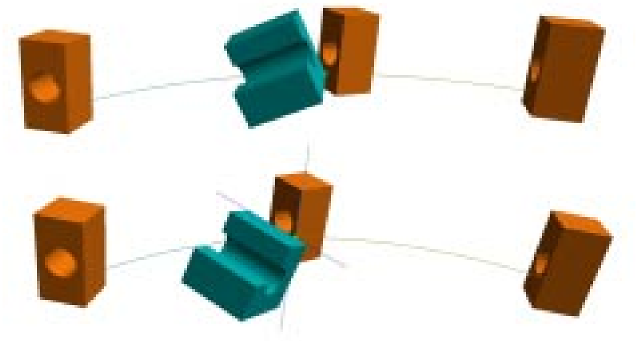

Figure 1: Collisions between a moving an object A, the block with a through hole, and a static object B, the L-shaped extrusion. A moves from right to left along a screw and its intermediate instance shows it at the moment of first collision with B for two different configurations: a face of A collides with a vertex of B (top) and an edge of A collides with an edge of B (bottom)

Introduction

The calculation of collisions amongst moving 3D objects and between moving objects and static obstacles has challenged animation experts and medical engineers for more than two decades [1, 7, 10, 38, 27, 32]. It has also been extensively studied in robotics [36]. We focus here on rigid bodies, and more precisely on polyhedra, and do not address collisions of deformable models [48, 2]. When complex 3D motions are involved, the poses (position and orientation) of each moving object are usually evaluated using a series of small time increments. At each stage of this simulation, the transformed instance of each object is tested against the instances of other objects using a static interference test [37]. Simple containing bounds [44], convex decompositions [3, 18, 19], hierarchical models [4, 9, 42, 22, 25, 26, 29, 35, 39, 41], and minimal distance computation [6, 15, 23, 37, 43] and tracking [13, 14, 5, 24] techniques have been used to reduce the frequency and the complexity of the interference tests [17, 20, 21, 31]. When an interference is detected, the last interval may be refined adaptively, until an accurate interval around the initial collision time is isolated. Even when instantaneous velocities or bounds on velocities are used to estimate the duration of collision-free intervals [11, 21], the number of interference test that are necessary to guarantee that all collisions are detected imposes limits on the performance of these collision detection approaches. By contrast to the above approaches, which are based on series of static interference tests, we propose a collision prediction approach, in which we compute the time and location of collisions directly from the relative motion of pairs of objects. Instead of computing swept volumes[36, 28, 34, 46, 49], or intersecting four-dimensional models swept by the moving objects in the space-time domain [12], we detect all occurrences of face/vertex and edge/edge collisions [16], and report the first one to occur. When both objects move along straight-line translations [10, 11] or both are rotating around the same axis [47], these direct collisions calculations may be reduced to the evaluation of a series of linear or quadratic expressions.

For more general motions where linear translation and rotation about different axes are allowed, trigonometric functions in the collision equation cannot be removed if the rotation angle and the translation displacement are interpolated as a linear function of a single parameter. In [17, 33, 47] these trigonometric functions were removed by nonlinear interpolation of the rotation angle, resulting in cubic or higher order polynomials. These nonlinear interpolations on angle can be considered to approximate linear ones, when the rotation angle is small. However, note that if the center of the rotation is far from the geometric center of the object, nonlinear interpolation on rotation produces a distorted trajectory, which may be quite different from the linear one. Thus, care must be taken in choosing the center of rotation.

Instead, to obtain a simple formulation of the exact collision parameters, we propose in this paper to approximate the relative motion[1] between any two objects by a continuous series of screw motion segments. For each screw motion segment, we compute the times of collision between all vertices of the first object and the faces of the other, between the faces of the first object and the vertices of the other, and between each edge of the first object and each edge of the second object. We report the smallest of these times. We assume of course that objects are initially disjoint.

Furthermore, to reduce the overall computational costs, we have developed fast interference tests between simple three-dimensional or parametric bounds. We use them to quickly reject most object pairs, vertex/face pairs, and edge/edge pairs that do not collide. The collision test and the computation of the time of first collision for each one of the remaining vertex/face and edge/edge pairs require solving a low-degree trigonometric equation in one variable. Once the minimal time of collision is found, we compute the collision point. Our direct collision computation techniques and simple rejection tests could of course be combined with hierarchical approaches that have been mentioned earlier. In the next section, we briefly explain how the screw motions are computed. Then, we present the simple rejection tests. Finally, we provide the details of the exact time to collision calculations and present implementation results.

Approximating screw motions

We propose to use a direct screw motion[2] to approximate the motion of an object A relative to a possibly moving object B. The screw motion interpolates the relative pose (position and orientation) of A at the beginning and ending of a given time interval, which we parameterize with t varying between 0 and 1. The reader further interested in screw theory and pose interpolation can refer to [28, 40, 50]. An interpolating, direct, screw motion is unique, except for the special case of 180° rotation between the starting and ending poses. By definition, it interpolates any two poses by a combination of a minimal angle rotation with a shortest vector translation. Note that the rotation axis is parallel to the translation vector. The interpolation is independent of the choice of coordinate system, thus the roles of A and B may be interchanged without affecting the results.

Because the maximum error[3] of a screw approximation of a motion segment is expensive to evaluate exactly for all points of an object, we advocate the following conservative estimation. We express the discrepancy motion, $D(t)$, as the product of the approximating screw motion by the inverse of the original motion. We then apply $D(t)$ to the eight vertices of an axis aligned [4] block containing A. The error is bounded by the maximum distance between the original position of each one of these vertices and its image by $D(t)$, as t varies between 0 and 1. Hence, we have reduced the error estimation problem to one of computing geometric bounds on the trajectory of several isolated points. When the error estimate exceeds a predetermined tolerance, the time interval is split in two and the motion split into two screw-motion segments continuously joined at the time where the two intervals meet.

The parameters of the direct screw motion may be computed easily for each segment. Assume that at instant t, the pose of object A is represented by a transformation that is the composition of a rotation $\mathbf{R}_A(t)$, which is represented by a 3×3 matrix, with a translation by a vector $\mathbf{v}_A(t)$. Assume that the pose of object B is similarly represented by a rotation $\mathbf{R}_B(t)$ and a translation $\mathbf{v}_B(t)$. We can express the relative motion of A with respect to B as the combination of a rotation $\mathbf{R}(t)$ with a translation by a vector $\mathbf{v}(t)$. $\mathbf{R}(t)$ may be represented by a 3×3 matrix that results from the multiplication of the transpose of the matrix of $\mathbf{R}_B(t)$ by the matrix of $\mathbf{R}_A(t)$. The vector $\mathbf{v}(t)$ is the difference between $\mathbf{v}_A(t)$, transformed by the transpose of the matrix of $\mathbf{R}_B(t)$, and $\mathbf{v}_B(t)$.

The parameters for a direct screw motion that interpolates the relative position of the two objects at times 0 and 1 may be simply derived from the matrix $\mathbf{R}^T(0)\mathbf{R}(1)$ and from the vector $\mathbf{v}(1)-\mathbf{v}(0)$. These parameters define the direction $\mathbf{s}$, a fixed point $\mathbf{p}$, a translation distance d along $\mathbf{s}$, and the angle b of a rotation around the axis parallel to $\mathbf{s}$ and passing through $\mathbf{p}$.

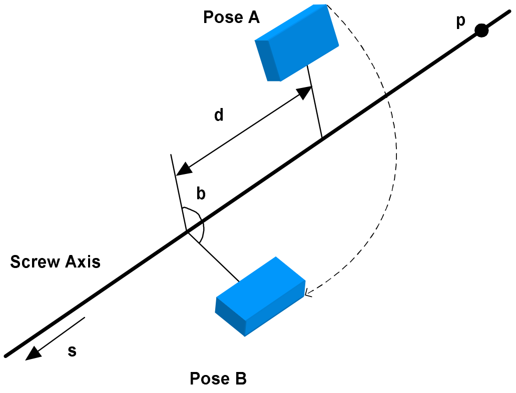

Figure 2: Screw parameters $\mathbf{s},\mathbf{p},d,b$

Computation of Screw Parameters $\mathbf{s}$, $\mathbf{p}$, d and b

Suppose that two poses of an object at time time $t=0$ to $t=1$ are given and we want to find a screw motion that interpolates them. In this section, we briefly discuss computing such screw parameters. Full description of computing pose interpolating screw pa-

[1] The ralative motion of an object A with respect to a moving object B is the motion of A in the body coordinate system of B

[2] A screw motion with samllest angle.

[3] The error is defined for each point of an object at a given time as the distances between its images in original and approximating poses.

[4] Aligned to the axes of global coordinate system.

rameters can be found in [45]. Let $\mathbf{R}_x(t), \mathbf{R}_y(t), \mathbf{R}_z(t)$ represent respectively the first, second, and third columns of $\mathbf{R}(t)$ and let $\Delta\mathbf{R}_{x,y,z} \equiv (\mathbf{R}_{x,y,z}(1) - \mathbf{R}_{x,y,z}(0))$. Fig. 3 illustrates these vectors. Then the rotation axis and rotation angle can be computed as

$$\mathbf{s} = \tilde{\mathbf{s}}/|\tilde{\mathbf{s}}| \ \text{,where } \tilde{\mathbf{s}} = \Delta\mathbf{R}_x \times \Delta\mathbf{R}_y + \Delta\mathbf{R}_y \times \Delta\mathbf{R}_z + \Delta\mathbf{R}_z \times \Delta\mathbf{R}_x$$
$$b = 2\sin^{-1}\frac{|\Delta\mathbf{R}_x|/2}{|\mathbf{s} \times \mathbf{R}_x(0)|} \qquad (1)$$

We have expressed $\tilde{\mathbf{s}}$ as the sum of three cross-products, so as to guarantee that the formula works in all situations and gives the maximum accuracy. Note however that at most one of $\Delta\mathbf{R}_x$, $\Delta\mathbf{R}_y$, or $\Delta\mathbf{R}_z$ can be null and therefore at most two of the cross-products can be null. Therefore, if $\Delta\mathbf{R}_x$ and $\Delta\mathbf{R}_y$ are both not null, we could also compute $\tilde{\mathbf{s}}$ as $\Delta\mathbf{R}_x \times \Delta\mathbf{R}_y$.

After computing the screw axis direction vector $\mathbf{s}$ and rotation angle b, a point on the screw axis $\mathbf{p}$ and translation d can be computed. Consider a point $\mathbf{o}$ in the object. Let $\mathbf{o}(t)$ be the position of $\mathbf{o}$ at time t. Since poses of time $t = 0$ and $t = 1$ are given, $\mathbf{o}(0)$ and $\mathbf{o}(1)$ are also given. From Fig. 3, d and $\mathbf{p}$ can be simply computed by

$$d = (\mathbf{o}(1) - \mathbf{o}(0)) \cdot \mathbf{s}$$
$$\mathbf{p} = \frac{1}{2}\left(\mathbf{o}(1) + \mathbf{o}(0) + \frac{\mathbf{s} \times (\mathbf{o}(1) - \mathbf{o}(0))}{\tan(b/2)}\right) \qquad (2)$$

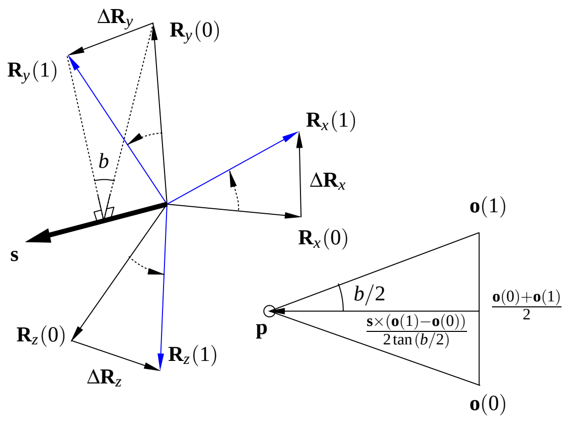

Figure 3: Computation of screw parameters $\mathbf{s}, \mathbf{p}, d, b$

Early Rejection Tests

In this section, we propose a series of simple tests for identifying situations where collision is clearly impossible. We assume that, in a direct pose interpolating screw motions, the magnitude of the rotation angle b is less than π, without loss of generality. When expressed in a relative coordinate of B, object A moves along a relative screw motion and object B is static. We assume for simplicity that the boundaries of both objects have been triangulated and that the objects are initially disjoint. (The latter conditions may be tested using a static interference test.) We must test for vertex/triangle, triangle/vertex, and edge/edge collisions. Since the triangle/vertex case can be converted to a vertex/triangle case by swapping the roles of A and B, or equivalently by inverting $\mathbf{s}$, there are only two different cases to consider. If the two objects have n and m vertices, $O(nm)$ pairs of entities need to be tested. We describe here quick rejection tests. They could be combined with hierarchy based algorithms, such as those for example based on sphere hierarchies [30].

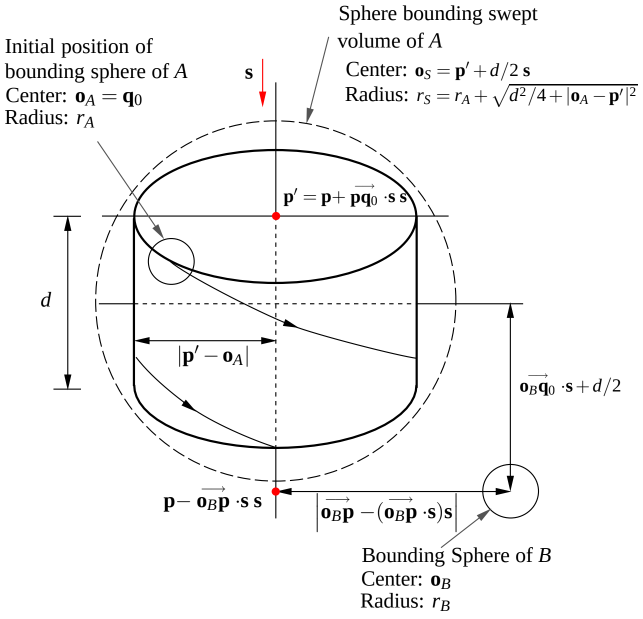
Sphere bounding swept volume of A
Center: $\mathbf{o}_S = \mathbf{p}' + d/2\,\mathbf{s}$
Radius: $r_S = r_A + \sqrt{d^2/4 + |\mathbf{o}_A - \mathbf{p}'|^2}$
Initial position of bounding sphere of A
Center: $\mathbf{o}_A = \mathbf{q}_0$
Radius: r_A
$\mathbf{s}$
$\mathbf{p}' = \mathbf{p} + \overrightarrow{\mathbf{p}\mathbf{q}_0} \cdot \mathbf{s}\,\mathbf{s}$
d
$|\mathbf{p}' - \mathbf{o}_A|$
$\overrightarrow{\mathbf{o}_B\mathbf{q}_0} \cdot \mathbf{s} + d/2$
$\mathbf{p} - \overrightarrow{\mathbf{o}_B\mathbf{p}} \cdot \mathbf{s}\,\mathbf{s}$
$|\overrightarrow{\mathbf{o}_B\mathbf{p}} - (\overrightarrow{\mathbf{o}_B\mathbf{p}} \cdot \mathbf{s})\mathbf{s}|$
Bounding Sphere of B
Center: $\mathbf{o}_B$
Radius: r_B

Figure 4: Trivial rejections: cylinder/sphere collision

Rejection Tests for Entire Objects

- We first check the collision between the bounding sphere, $Sph(\mathbf{o}_B, r_B)$, of B and the sphere, $Sph(\mathbf{o}_S, r_S)$, that encloses the volume swept by a bounding sphere of A. The symbols $\mathbf{o}$ and r with the appropriate subscripts represent respectively the centers and radii of these spheres. As is illustrated in Fig. 4, center and radius of $Sph(\mathbf{o}_S, r_S)$ are computed as $\mathbf{o}_S = \mathbf{p}' + d/2\,\mathbf{s}$ and $r_S = r_A + \sqrt{d^2/4 + |\mathbf{o}_A - \mathbf{p}'|^2}$. The rejection condition is $|\mathbf{o}_s - \mathbf{o}_B| > r_B + r_S$

- If the two bounding spheres are intersecting, we check whether the sphere around B and the infinite cylinder centered around the axis of the screw and containing the $Sph(\mathbf{o}_A, r_A)$ intersect. First, the distance between the screw axis and the center of the bounding sphere of B is computed. The rejection condition is

$$|\overrightarrow{\mathbf{o}_B\mathbf{p}} - \overrightarrow{\mathbf{o}_B\mathbf{p}} \cdot \mathbf{s}\,\mathbf{s}| > |\mathbf{o}_A - \mathbf{p}'| + r_A + r_B \qquad (3)$$

  If $|\mathbf{p}' - \mathbf{o}_A|$ is larger than r_A, the following condition can also be used as a rejection test, identifying the cases where B lies inside a shrunken version of the cylinder.

$$|\overrightarrow{\mathbf{o}_B\mathbf{p}} - \overrightarrow{\mathbf{o}_B\mathbf{p}} \cdot \mathbf{s}\,\mathbf{s}| < |\mathbf{o}_A - \mathbf{p}'| - r_A - r_B \qquad (4)$$

- Finally, the planes that cap the finite cylinder that contains the volume swept by the sphere around A are considered. If $|(\mathbf{o}_B - \mathbf{o}_S) \cdot \mathbf{s}| > d/2 + r_A + r_B$, there is no intersection.

Vertex/Triangle rejection tests

If the above tests fail to reject the pair of objects, we test for collision between the individual pairs of elements of their boundaries. First we test all vertex/triangle and triangle/vertex pairs. As depicted in Fig. 5, the following rejection situations for a vertex and triangle pair are considered.

- After projecting the vertices of a triangle to the screw axis, the distances to the cylinder that contains the helix along

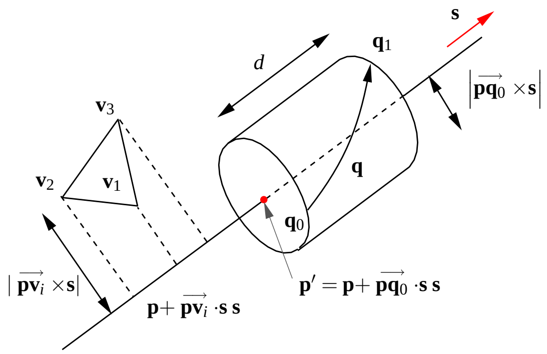
$\mathbf{s}$
$\mathbf{q}_1$
d
$|\overrightarrow{\mathbf{pq}_0} \times \mathbf{s}|$
$\mathbf{v}_3$
$\mathbf{v}_2$
$\mathbf{v}_1$
$\mathbf{q}$
$\mathbf{q}_0$
$|\overrightarrow{\mathbf{pv}_i} \times \mathbf{s}|$
$\mathbf{p}+\overrightarrow{\mathbf{pv}_i} \cdot \mathbf{s}\,\mathbf{s}$
$\mathbf{p}' = \mathbf{p}+\overrightarrow{\mathbf{pq}_0} \cdot \mathbf{s}\,\mathbf{s}$

Figure 5: Trivial rejections: vertex/triangle

which $\mathbf{q}$ moves are tested. If $(\mathbf{q}_0 - \mathbf{v}_i) \cdot \mathbf{s} > 0, \forall i = 1,2,3$ or $(\mathbf{q}_0 - \mathbf{v}_i) \cdot \mathbf{s} < -d, \forall i = 1,2,3$, then there is no collision.

- Rejection is also possible when the triangle lies inside the cylinder. If $|(\mathbf{v}_i - \mathbf{p}) \cdot \mathbf{s}| < |(\mathbf{q}_0 - \mathbf{p}) \cdot \mathbf{s}|, \forall i = 1,2,3$, then there is no collision.

- Trivial rejection is also possible when the triangle is out of the angle swept by the vertex as projected on a plane perpendicular to the screw axis. From the plane equations that include the screw axis, $\mathbf{q}_0$ and $\mathbf{q}_1$, if $0 \leq b \leq \pi, (\mathbf{q}_0 - \mathbf{p}) \times \mathbf{s} \cdot (\mathbf{v}_i - \mathbf{p}) < 0, \forall i = 1,2,3$ or $(\mathbf{q}_1 - \mathbf{p}) \times \mathbf{s} \cdot (\mathbf{v}_i - \mathbf{p}) < 0, \forall i = 1,2,3$, there is no collision. If $\pi < b < 2\pi$, the six conditions must be satisfied at the same time, *i.e.*, if $(\mathbf{q}_0 - \mathbf{p}) \times \mathbf{s} \cdot (\mathbf{v}_i - \mathbf{p}) < 0$ and $(\mathbf{q}_1 - \mathbf{p}) \times \mathbf{s} \cdot (\mathbf{v}_i - \mathbf{p}) > 0, \forall i = 1,2,3$, there is no collision.

Edge/edge rejection tests

Finally, we develop rejection tests for edge/edge pairs. Assume that an edge of A having vertices $\mathbf{a}_1$ and $\mathbf{a}_2$ is moving along a screw segment. Let $\mathbf{b}_1$ and $\mathbf{b}_2$ be vertices of an edge of static object B. For convenience, define the two radii as $r_{\min} = \min((\mathbf{a}_{1,2} - \mathbf{p}) \cdot \mathbf{s})$ and $r_{\max} = \max((\mathbf{a}_{1,2} - \mathbf{p}) \cdot \mathbf{s})$.

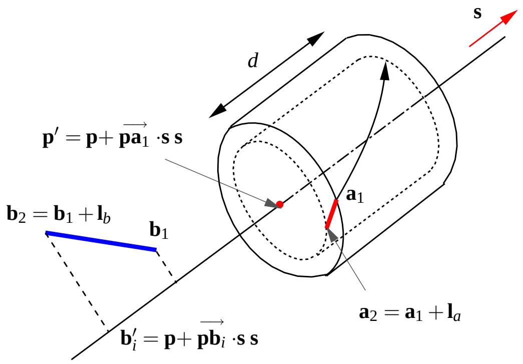
$\mathbf{s}$
d
$\mathbf{p}' = \mathbf{p}+\overrightarrow{\mathbf{pa}_1} \cdot \mathbf{s}\,\mathbf{s}$
$\mathbf{b}_2 = \mathbf{b}_1 + \mathbf{l}_b$
$\mathbf{b}_1$
$\mathbf{a}_1$
$\mathbf{a}_2 = \mathbf{a}_1 + \mathbf{l}_a$
$\mathbf{b}_i' = \mathbf{p}+\overrightarrow{\mathbf{pb}_i} \cdot \mathbf{s}\,\mathbf{s}$

Figure 6: Trivial rejections: edge/edge

- If the projected line segment does not intersect with the cylindrical annulus, there is no collision. The rejection condition is $(\mathbf{a}_{1,2} - \mathbf{b}_{1,2}) \cdot \mathbf{s} > 0$ or $(\mathbf{a}_{1,2} - \mathbf{b}_{1,2}) \cdot \mathbf{s} < -d$.

- When the edge $\overline{\mathbf{b}_1\mathbf{b}_2}$ is inside the annulus, there is no collision. The rejection condition is $|(\mathbf{b}_{1,2} - \mathbf{p}) \cdot \mathbf{s}| < r_{\min}$.

- When the minimum distance between the edge $\overline{\mathbf{b}_1\mathbf{b}_2}$ and the screw axis is larger than $r_{\max}$, there is no collision. Let $\mathbf{n} = \mathbf{s} \times \mathbf{l}_b / |\mathbf{s} \times \mathbf{l}_b|$ be the common normal of the screw axis and $\overline{\mathbf{b}_1\mathbf{b}_2}$. Then the minimum distance is γ for some t and β that can satisfy the equation $\mathbf{p} + t\mathbf{s} + \mathbf{n}\gamma = \mathbf{b}1 + \mathbf{l}_b\beta$. The reject condition is $|\gamma| = |(\mathbf{b}_1 - \mathbf{p}) \times \mathbf{s} \cdot \mathbf{l}_b / (\mathbf{n} \times \mathbf{s} \cdot \mathbf{l}_b))| > r_{\max}$.

Collision Time and Contact Calculation

As justified above, when the trivial rejection test fail, we need to test for the following collision cases: vertex/triangle and edge/edge. For each case, the collision condition is formulated as a function of time and is solved numerically using a Newton's iterative method with the careful choices of initial guesses.

Screw Trajectory in Analytic Form

Each point of an object undergoing a screw motion moves along a helix around the screw axis, which contains a fixed point $\mathbf{p}$ and has the direction $\mathbf{s}$ as illustrated in Fig. 7. Our deterministic collision prediction is based on the Rodrigues equations [8]. Let $\mathbf{q}_0$ be the initial position and $\mathbf{q}(t)$ be the position at time t. Let $\mathbf{p}'$ be an orthogonal projection of $\mathbf{q}_0$ onto the screw axis and $\mathbf{q}'$ be $\mathbf{q}_0$ rotated about $\mathbf{s}$ by the angle tb. We first derive $\overrightarrow{\mathbf{p}'\mathbf{q}'} = \mathbf{q}' - \mathbf{p}'$, which is the rotational motion of vector from $\mathbf{p}'$ to $\mathbf{q}_0$ about $\mathbf{s}$ by the angle tb. Assuming $|\mathbf{s}| = 1$, from Fig. 7,

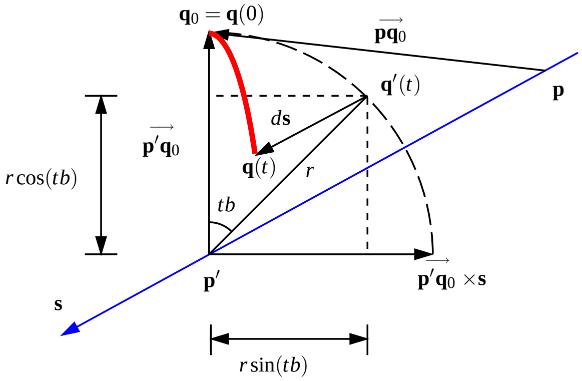
$\mathbf{q}_0 = \mathbf{q}(0)$
$\overrightarrow{\mathbf{pq}_0}$
$\mathbf{q}'(t)$
$\mathbf{p}$
$\overrightarrow{\mathbf{p}'\mathbf{q}_0}$
$d\mathbf{s}$
$\mathbf{q}(t)$
r
$r\cos(tb)$
tb
$\mathbf{p}'$
$\overrightarrow{\mathbf{p}'\mathbf{q}_0} \times \mathbf{s}$
$\mathbf{s}$
$r\sin(tb)$

Figure 7: Rotation and projection of a point $\mathbf{q}_0$

$$\begin{aligned}\overrightarrow{\mathbf{p}'\mathbf{q}'} = \mathbf{q}'(t) - \mathbf{p}' &= r\cos(tb)\frac{\overrightarrow{\mathbf{p}'\mathbf{q}_0}}{r} + r\sin(tb)\frac{\overrightarrow{\mathbf{p}'\mathbf{q}_0} \times \mathbf{s}}{r} \\ &= \cos(tb)\,\overrightarrow{\mathbf{p}'\mathbf{q}_0} + \sin(tb)\,\overrightarrow{\mathbf{p}'\mathbf{q}_0} \times \mathbf{s}\end{aligned} \tag{5}$$

The vector from $\mathbf{p}$ to $\mathbf{q}'(t)$ can be computed as

$$\begin{aligned}\overrightarrow{\mathbf{pq}'} &= \overrightarrow{\mathbf{p}'\mathbf{q}'} + (\overrightarrow{\mathbf{pq}_0} \cdot \mathbf{s})\mathbf{s} \\ &= \cos(tb)\,\overrightarrow{\mathbf{pq}_0} + \sin(tb)\,\overrightarrow{\mathbf{pq}_0} \times \mathbf{s} + (1 - \cos(tb))(\overrightarrow{\mathbf{pq}_0} \cdot \mathbf{s})\mathbf{s} \\ &\equiv \vec{\phi}(\overrightarrow{\mathbf{pq}_0}, tb)\end{aligned} \tag{6}$$

where we defined a new notation $\vec{\phi}(\mathbf{a}, tb)$ as a vector $\mathbf{a}$ rotated about the screw axis by the angle tb. From (5), $\mathbf{q}'(t) = \mathbf{p} + \overrightarrow{\mathbf{pq}'} = \mathbf{p} + \vec{\phi}(\overrightarrow{\mathbf{pq}_0}, tb)$. The screw trajectory $\mathbf{q}(t)$ can be obtained by adding the translation vector $td\mathbf{s}$.

$$\mathbf{q}(t) = td\mathbf{s} + \mathbf{q}'(t) \;\; = \;\; \mathbf{p} + td\mathbf{s} + \vec{\phi}(\overrightarrow{\mathbf{pq}_0}, tb) \tag{7}$$

Vertex/Triangle Case

The plane that contains a triangle is defined by a unit normal $\mathbf{n}_s$ and a signed distance d_s from the origin. Any point $\mathbf{q}$ in this plane equation satisfies $d_s + \mathbf{q} \cdot \mathbf{n}_s = 0$. The intersection between this plane and the screw trajectory can be calculated by substituting equation (7) into this plane equation.

$$0 = d_s + \mathbf{p} \cdot \mathbf{n}_s + td\,(\mathbf{s} \cdot \mathbf{n}_s) + \vec{\phi}(\overrightarrow{\mathbf{pq}_0}, tb) \cdot \mathbf{n}_s \tag{8}$$

We can rewrite (8) in the following form

$$\begin{aligned} f(t) &= c_0 + c_1 t + c_2 \cos(tb) + c_3 \sin(tb) \\ &= c_0 + c_1 t + A\cos(tb - \theta) \end{aligned} \tag{9}$$

Where the coefficients are given by [5]

$$\begin{aligned} &c_0 = \mathbf{p} \cdot \mathbf{n}_s + (\overrightarrow{\mathbf{pq}_0} \cdot \mathbf{s})(\mathbf{s} \cdot \mathbf{n}_s) + d_s \quad , \ c_1 = d\, \mathbf{s} \cdot \mathbf{n}_s \\ &c_2 = \overrightarrow{\mathbf{pq}_0} \cdot \mathbf{n}_s + (\overrightarrow{\mathbf{pq}_0} \cdot \mathbf{s})(\mathbf{s} \cdot \mathbf{n}_s) \quad , \ c_3 = (\overrightarrow{\mathbf{pq}_0} \times \mathbf{s}) \cdot \mathbf{n}_s \\ &A = \sqrt{c_2^2 + c_3^2} \quad , \ \theta = \tan_2^{-1}(c_3, c_2) \end{aligned} \tag{11}$$

Since $f(t)$ does not have an analytic solution, we use Newton iterations. Consider extremal points of $f(t)$ where $f'(t) = 0$. Since $f(t)$ is monotonic between these points, if the signs of at two consecutive extremal points are identical, there is no solution between them. If the signs are different, there is one solution. The extremal points computed from the condition $f'(t) = c_1 - Ab\sin(bt - \theta) = 0$ are

$$t = (\theta + (-1)^m \alpha + m\pi)/b\,, \ \ \alpha = \sin^{-1}\left(\frac{c_1}{Ab}\right) \tag{12}$$

where m is integer. Define m_1 to be the smallest m such that $t > t_0$. Then m_1 can be calculated as

$$m_1 = \min\left(2\left\lceil \frac{-\theta - \alpha + bt_0}{2\pi} \right\rceil, 2\left\lceil \frac{-\theta + \alpha + bt_0}{2\pi} - \frac{1}{2} \right\rceil\right) \tag{13}$$

where it is assumed that $b > 0$ without loss of generality. Once we find m_1, we can increase it by one to calculate the next extremal point and then compare the signs of $f(t)$ at these two points. If the signs are different there is one solution and we need to perform Newton iterations starting with the reflection points as the initial guess, which can be computed from $f''(t) = -Ab^2\cos(bt - \theta) = 0$ that yields $bt - \theta = n\pi + \pi/2$ where n is integer. We have the reflection points at

$$t = \frac{\theta + n\pi + \pi/2}{b} \tag{14}$$

We start our root-finding process with n_1, the smallest n such that $t \geq 0$, which is computed as

$$n_1 = \left\lceil -\frac{1}{2} - \frac{\theta}{\pi} \right\rceil \tag{15}$$

Note that in pose interpolating screw motion, $b \in [-\pi, \pi]$ and there could be up to two reflection points in the interval and we need to start from these two points. Also, note that Newton iteration in general converges in three to four iterations.

Whenever a new helix/plane intersection point is found, a triangle containnment test is performed to ensure that the vertex actually collides with the triangle. If the intersection lies inside the triangle, the search is stopped. Fig. 8 and 9 show the trajectory and collision points in 3D on the right and the plot of the $f(t)$ with its two roots in the left. Once the solution is found, only the roots that pass the the triangle inside test are accepted. If more than two roots passes the inside test, we choose the one with smallest t since only the first collision is interesting.

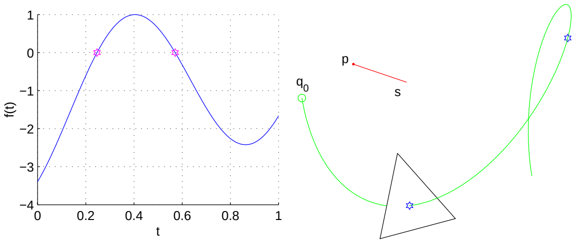

Figure 8: Vertex/Triangle collision.

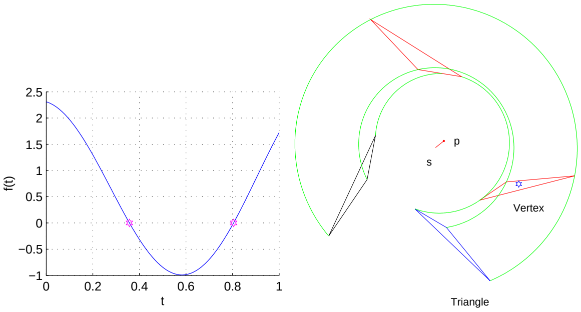

Figure 9: Triangle/Vertex collision.

Edge/Edge Collision Case

Let $(\mathbf{a}_1, \mathbf{a}_2)$ be an edge of A and $(\mathbf{b}_1, \mathbf{b}_2)$ be an edge of B. Let $\mathbf{l}_a = \mathbf{a}_2 - \mathbf{a}_1$ and $\mathbf{l}_b = \mathbf{b}_2 - \mathbf{b}_1$. We express a point on the edge of A as $\mathbf{a}_1 + \alpha \mathbf{l}_a$ and a point on the edge of B as $\mathbf{b}_1 + \beta \mathbf{l}_b$, where $0 \leq \alpha, \beta \leq 1$. Let $\mathbf{q}_\alpha(t, \alpha)$ be the point $\mathbf{a}_1 + \alpha \mathbf{l}_a$ transformed by the screw motion at time t. Then, the collision detection problem amounts to finding three variables t, α and β for which $\mathbf{q}_\alpha(t, \alpha) = \mathbf{b}_1 + \beta \mathbf{l}_b$. $\mathbf{q}_\alpha$ can be calculated by substituting $\mathbf{a}_1 + \alpha \mathbf{l}_a$ into (7),

$$\mathbf{q}_\alpha(t, \alpha) = \mathbf{q}_a(t) + \alpha \vec{\phi}(\mathbf{l}_a, tb) \tag{16}$$

where $\mathbf{q}_a(t)$ and $\vec{\phi}(\mathbf{l}_a, tb)$ are defined as

$$\begin{aligned} \mathbf{q}_a(t) &= \mathbf{p} + td\mathbf{s} + \vec{\phi}(\mathbf{a}_1, tb) \\ \vec{\phi}(\mathbf{l}_a, tb) &= \cos(tb)\mathbf{l}_a + \sin(tb)\mathbf{l}_a \times \mathbf{s} + (1 - \cos(tb))\mathbf{l}_a \cdot \mathbf{s}\,\mathbf{s} \end{aligned} \tag{17}$$

Note that $\vec{\phi}(\mathbf{l}_a, tb)$ is the transformed $\mathbf{l}_a$ at time t. The collision condition is

$$\mathbf{q}_\alpha(t, \alpha) = \mathbf{q}_a(t) + \alpha \vec{\phi}(\mathbf{l}_a, tb) = b + \beta \mathbf{l}_b \tag{18}$$

To eliminate α, we take the cross product with $\vec{\phi}(\mathbf{l}_a, tb)$, yielding $\mathbf{q}_a \times \vec{\phi}(\mathbf{l}_a, tb) = b \times \vec{\phi}(\mathbf{l}_a, tb) + \beta \mathbf{l}_b \times \vec{\phi}(\mathbf{l}_a, tb)$ and then we take the dot product with $\mathbf{l}_b$ resulting in $\mathbf{q}_\alpha \times \vec{\phi}(\mathbf{l}_a, tb) \cdot \mathbf{l}_b = b \times \vec{\phi}(\mathbf{l}_a, tb) \cdot \mathbf{l}_b$. Using the vector identity $\mathbf{u} \times \mathbf{v} \cdot \mathbf{w} = \mathbf{u} \cdot \mathbf{v} \times \mathbf{w}$, this can be written as

$$f(t) \equiv (\mathbf{q}_a - \mathbf{b}_1) \cdot (\vec{\phi}(\mathbf{l}_a, tb) \times \mathbf{l}_b) = 0 \tag{19}$$

[5] These coefficients can also be derived in the local coordinate system $(\mathbf{i}, \mathbf{j}, \mathbf{k}, \mathbf{o})$ of the screw, where $\mathbf{o} = \mathbf{p}', \mathbf{k} = \mathbf{s}$, and $\mathbf{i}$ is parallel to $\mathbf{p}'\mathbf{q}_0$. The screw trajectory is $\mathbf{q}(t) = |\mathbf{q}_0|\cos(tb)\mathbf{i} + |\mathbf{q}_0|\sin(tb)\mathbf{j} + td\mathbf{k}$. The screw intersects plane $d_s + \mathbf{q}\mathbf{n}_s = 0$ for values of t satisfying

$$d_s + (|\mathbf{q}_0|\cos(tb), |\mathbf{q}_0|\sin(tb), td) \cdot \mathbf{n}_s = 0 \tag{10}$$

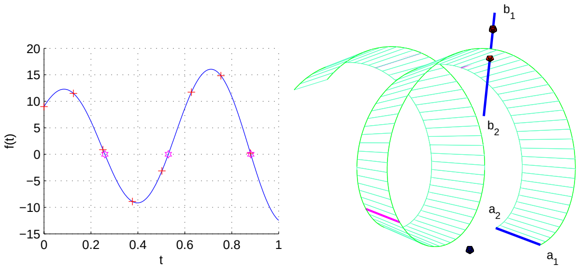

Figure 10: Edge/Edge collision.

Using vector identities[6] and assuming $|\mathbf{s}| = 1$, $f(t)$ can be simplified to the following form

$$f(t) = c_0 + (c_1 + c_3 t)\cos(tb) + (c_2 + c_4 t)\sin(tb) \qquad (20)$$

with the coefficients [7]

$$\begin{aligned}
c_0 &= \overrightarrow{\mathbf{pb}_1} \cdot (\mathbf{l}_b \times \mathbf{s})(\mathbf{l}_a \cdot \mathbf{s}) + \overrightarrow{\mathbf{pa}_1} \cdot (\mathbf{l}_a \times \mathbf{s})(\mathbf{l}_b \cdot \mathbf{s}) \\
c_1 &= (\overrightarrow{\mathbf{pa}_1} \cdot \mathbf{s}\, \mathbf{s} - \overrightarrow{\mathbf{pb}_1}) \cdot (\mathbf{l}_a \times \mathbf{l}_b) - (\overrightarrow{\mathbf{pa}_1} + \overrightarrow{\mathbf{pb}_1}) \cdot (\mathbf{l}_b \times \mathbf{s})(\mathbf{l}_a \cdot \mathbf{s}) \\
c_2 &= -\overrightarrow{\mathbf{pa}_1} \cdot (\mathbf{l}_a \times (\mathbf{l}_b \times \mathbf{s})) + \overrightarrow{\mathbf{pb}_1} \cdot (\mathbf{l}_b \times (\mathbf{l}_a \times \mathbf{s})) \\
c_3 &= d\, \mathbf{s} \cdot \mathbf{l}_a \times \mathbf{l}_b \\
c_4 &= d\, (\mathbf{l}_a \times \mathbf{s}) \cdot (\mathbf{l}_b \times \mathbf{s})
\end{aligned} \qquad (21)$$

To find the solution, we perform Newton iterations with initial guesses that uniformly sample the search interval at a frequency five times larger than $b/2\pi$, the period in $f(t)$. Note that since $b \in [-\pi, \pi]$ for pose interpolating screw motion, we may need up to three initial guesses. Once a solution is found, α and β can be computed as

$$\begin{aligned}
\alpha &= \frac{((\mathbf{b}_1 - \mathbf{q}_a) \times \mathbf{l}_b) \cdot \left(\vec{\phi}(\mathbf{l}_a, tb) \times \mathbf{l}_b\right)}{|\vec{\phi}(\mathbf{l}_a, tb) \times \mathbf{l}_b|^2} \\
\beta &= \frac{\left(\mathbf{q}_a + \alpha\vec{\phi}(\mathbf{l}_a, tb) - \mathbf{b}_1\right) \cdot \mathbf{l}_b}{|\mathbf{l}_b|^2}
\end{aligned} \qquad (22)$$

We are only interested in solutions where $t, \alpha, \beta \in [0,1]$. Note that singular situations such as zero length edge or parallel edge may be ignored since they are detected by vertex/triangle collision. Fig. 10 shows an example of a collision between two edges. In the left figure, red pluses are the initial guesses and red hexagons are the solutions found. Among these solutions, only the ones that satisfy $0 \leq \alpha, \beta \leq 1$ are accepted as valid contacts. Finally, the one with smallest t is reported as the collision time.

Results

To report the execution times and the benefits of our rejection tests, 50,000 random cases were generated and tested on an Pentium4 2.5GHz PC. The two polyhedra shown in Fig. 1 were used. The object A is placed at the origin at $t = 0$ and moves along a fixed screw trajectory with parameters $\mathbf{p} = [3,3,3]$, $d = 6$, $b = 2\pi$, and $\mathbf{s} = [1,1,1]/\sqrt{3}$. The object B is placed in arbitrary orientations and positions in the cube of size 15, centered at the origin, but outside of a cube of size 5 centered at the origin, to avoid initial interference with A. A has 80 vertices, 156 triangles and 234 edges. B has 85 vertices, 166 triangles and 249 edges. Therefore, there are 13,280 vertex/triangle, 13,260 triangle/vertex and 58,266 edge/edge pairs. If we treat the triangle/vertex case as vertex/triangle, there are 26,540 triangle/vertex pairs.

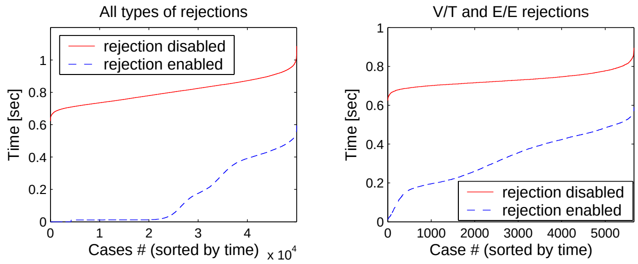

Figure 11: Left: with all types of rejections when objects may or mat not collide. Right: with vertex/triangle and edge/edge rejections when objects collide.

The left plot in Fig. 11 shows the timings for all colliding and non-colliding cases including the sphere/cylinder rejection. This results show that 50% of the cases were rejected by our cylinder/sphere object-pair rejection test. The execution time of these cylinder/sphere test is negligible. Right plot in Fig. 11 is results only for the cases when a collision actually occurred between the two objects. In these cases, 50.5% of the vertex/triangle and 66% of the edge/edge tests were rejected thanks to the corresponding rejection tests discussed earlier. The early rejection results in a speedup of 55%. The average execution time for rejection tests on 26,540 veltex/triangle and 58,266 edge/edge pairs was 32.8ms, which yields about 0.39μs for a single pair. If all the rejection tests failed, Newton iteration is performed. The execution time for computing an exact contact points by root finding is 9.5 μs in average.

Conclusions

The closed form expression of the trajectory of a point under a screw motion is used to predict the collision time and contact point between two polyhedra, whose relative motion is approximated by one or several screw motion segments. Univariate equations for collision conditions are derived for the vertex/triangle and the edge/edge collision cases. Several trivial rejection cases are proposed that exploit the properties of screw motions. Our tests show that early rejection tests speed up the computation by 55% in cases where a collision occurs.

REFERENCES

[1] N. Ahuja, R. T. Chien, R. Yen, and N. Bridwell. Interference detection and collision avoidance among three dimensional objects. *In I Annual National Conference on AI, Stanford University*, pages 44–48, August 1980.

[2] A. H. B. B. Von Herzen and H. R. Zatz. Geometric collisions for time-dependent parametric surfaces. *ACM Computer Graphics*, 24(4):39–48, August 1990.

[3] C. Bajaj and T. Dey. Convex decomposition of polyhedra and robustness. *SIAM Journal on Computing*, 21(2):9–64, April 1992.

[4] S. Bandi and D. Thalmann. An adaptive spatial subdivision of the object space for fast collision detection of animating rigid bodies. In *Proceedings of Eurographics '95*, pages 259–270, August 1995.

[6] $\mathbf{u} \cdot (\mathbf{v} \times \mathbf{w}) = (\mathbf{u} \times \mathbf{v}) \cdot \mathbf{w}$, $\mathbf{u} \times (\mathbf{v} \times \mathbf{w}) = (\mathbf{u} \cdot \mathbf{w})\mathbf{v} - (\mathbf{u} \cdot \mathbf{v})\mathbf{w}$, and $(\mathbf{u} \times \mathbf{v}) \cdot (\mathbf{u} \times \mathbf{w}) = \mathbf{v} \cdot \mathbf{w} - (\mathbf{u} \cdot \mathbf{v})(\mathbf{u} \cdot \mathbf{w})$

[7] Similarly to (10), these coefficient can be computed after the relevant transformation that aligns $\mathbf{s}$ with z axis and $\mathbf{a}_1$ onto x axis. Note that $\mathbf{q}_a$ and $\vec{\phi}(\mathbf{l}_a, tb)$ are $\mathbf{a}_1$ and $\mathbf{l}_a$ at time t.

[5] G. V. D. Bergen. A fast and robust gjk implementation for collision detection of convex objects. *Journal of Graphics*, 4(2):7–25, 1999.

[6] J. E. Bobrow. A direct optimization approach for obtaining the distance between convex polyhedra. *The International Journal of Robotics Research*, 8(3):65–76, June 1983.

[7] S. Bonner and R. B. Kelley. A representation scheme for rapid 3-d collision detection. In *Proceedings of IEEE International Symposium on Intelligent Control*, pages 320–325, Aug 1988.

[8] O. Bottema and B. Roth. *Theoretical Kinematics*. North-Holland Publishing Company, Amsterdam, New York, Oxford, pages 56–62, 1979.

[9] W. Bouma and G. Vanecek. Collision detection and analysis in a physical based simulation. In *Eurographics Workshop on Animation and Simulation*, pages 191–203, September 1991.

[10] J. W. Boyse. Interference detection among solids and surfaces. *Communications of ACM*, 22(1):3–9, January 1979.

[11] S. A. Cameron. A study of the clash detection problem in robotics. In *Proceedings of IEEE International Conference on Robotics and Automation*, pages 488–493, March 1985.

[12] S. A. Cameron. Collision detection by four-dimensional intersection testing. *IEEE Transactions on Robotics and Automation*, 6(3):291–302, June 1990.

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

[14] S. A. Cameron. Enhancing gjk: Computing minimum and penetration distances between convex polyhedra. In *Proceedings of IEEE International Conference on Robotics and Automation*, pages 3112–3117, April 1997.

[15] S. A. Cameron and R. K. Culley. Determining the minimum translational distance between two convex polyhedra. In *Proceedings of IEEE International Conference on Robotics and Automation*, pages 591–596, April 1986.

[16] J. Canny. *The Complexity of Robot Motion Planning*. MIT Press, Cambridge, MA, 1987.

[17] J. F. Canny. Collision detection for moving polyhedra. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 8(2):200–209, March 1986.

[18] B. Chazelle. Convex partitions of polyhedra: A lower bound and a worst-case optimal algorithm. *SIAM Journal on Computing*, 13:488–507, 1984.

[19] B. Chazelle, D. Dobkin, N. Shouraboura, and A. Tal. Strategies for polyhedral surface decomposition: an experimental study. *Computational Geometry: Theory and Applications*, 7(4-5):327–342, 484, 1997.

[20] J. D. Cohen, M. C. Lin, D. Manocha, and M. K. Ponamgi. I-collide: An interactive and exact collision detection system for large-scale environments. In *Proceedings of ACM International 3D Graphics Conference*, volume 1, pages 189–196, 1995.

[21] R. K. Culley and K. G. Kempf. A collision detection algorithm based on velocity and distance bounds. In *IEEE International Conference on Robotics and Automation*, pages 1064–1069, April 1986.

[22] D. Dobkin and D. Kirkpatrick. Determining the separation of preprocessed polyhedra - a unified approach. In *Lecture Notes in Computer Science*, volume 443, pages 400–413, JULY 1990.

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

[24] E. G. Gilbert, D. W. Johnson, and S. Keerthi. A fast procedure for computing the distance between complex objects in three dimensional space. *IEEE Journal of Robotics and Automattion*, 4(2):193–203, April 1988.

[25] S. Gottschalk, M. C. Lin, and D. Manocha. Obb-tree: A hierarchical structure for rapid interference detection. In *Proceedings of ACM Siggraph*, 1996.

[26] K. Hamada and Y. Hori. Octree-based approach to real-time collision-free path planning for robot manipulator. In *ACM96-MIE*, pages 705–710, 1996.

[27] V. Hayward. Fast collision detection scheme by recursive decomposition of a manipulator workspace. In *Proceedings of IEEE International Conference on Robotics and Automation*, pages 1044–1049, 1986.

[28] Z. Hu and Z. Ling. Generating swept volumes with instantaneous screw axes. In *Proceedings of 94 ASME Design Technical Conference, Part 1*, pages 7–14, 1994.

[29] P. Hubbard. Interactive collision detection. In *Proceedings of IEEE Symposium on Research Frontiers in Virtual Reality*, pages 24–31, 1993.

[30] P. M. Hubbard. Approximating polyhedra with spheres for time-critical collision detection. *ACM Transactions on Graphics*, 15(3):179–210, 1996.

[31] T. C. Hudson, M. C. Lin, J. D. Cohen, S. Gottschalk, and D. Manocha. V-collide: Accelerated collision detection for vrml. In *Proceedings of VRML*, 1997.

[32] P. Jimenez, F. Thomas, and C. Torras. 3d collision detection: a survey. *Computers and Graphics*, 25(2), 2001.

[33] P. Jimenez and C. Torras. Collision detection: a geometric approach. In *Modelling and Planning for Sensor Based Intelligent Robot Systems*, pages 68–85. World Scientific Pub. Co., 1995.

[34] J. Keiffe and L. Litvin. Swept volume determination and interference of moving 3-d solids. *ASME Journal of Mechanical Design*, 113:456–463, 1991.

[35] J. Klosowski, M. Held, J. Mitchell, H. Sowizral, and K. Zikan. Efficient collision detection using bounding volume hierarchies of k-dops. *IEEE Transactions on Visualization and Computer Graphics*, 4(1), 1998.

[36] J. U. Korein. *A Geometric Investigation of Reach*. The MIT Press, 1984.

[37] M. C. Lin and J. F. Canny. A fast algorithm for incremental distance calculation. In *Proceedings of IEEE International Conference on Robotics and Automation*, volume 2, pages 1008–1014, 1991.

[38] M. C. Lin and S. Gottschalk. Collision detection between geometric models: a survey. In *In Proceedings of IMA Conference on Mathematics of Surfaces*, volume 1, pages 602–608, May 1998.

[39] B. Martinez, A. P. DelPobil, and M. Perez. Very fast collision detection for practical motion planning. part i: The spatial representation. In *Proceedings of IEEE International Conference on Robotics and Automation*, pages 624–629, May 1998.

[40] M. Ohwovoriole and B. Roth. An extension of screw theory. *Transaction of ASME Journal of Mechanical Design*, 103:725–735, 1981.

[41] I. J. Palmer and R. L. Grimsdale. Collision detection for animation using sphere-trees. *Computer Graphics Forum*, 14(2):105–116, 1995.

[42] A. P. D. Pobil, M. A. Serna, and J. Llovet. A new representation for collision a voidance and detection. In *Proceedings of IEEE International Conference on Robotics and Automation*, volume 1, pages 246–251, May 1992.

[43] S. Quinlan. Efficient distance computation between non-convex objects. In *Proceedings of IEEE International Conference on Robotics and Automation*, volume 4, pages 3324–3329, 1994.

[44] E. Rimon and S. P. Boyd. Obstacle collision detection using best ellipsoid fit. *Journal of Intelligent and Robotic Systems*, 18:105–126, 1997.

[45] J. R. Rossignac and J. J. Kim. Computing and visualizing pose-interpolating 3d motions. *Computer Aided Design*, 33(4):279–291, April 2001.

[46] K. Sambandan and K. K. Wang. Five-axis swept volumes for graphic nc simulation and verification. *ASME DE*, 19(1):143–150, 1989.

[47] E. Schomer and C. Thiel. Efficient collision detection for moving polyhedra. In *Proceedings of the eleventh annual symposium on computational geometry*, pages 51–60, 1995.

[48] J. M. Snyder, A. R. Woodbury, K. Fleischer, B. Currin, and A. H. Barr. Interval methods for multi-point collisions between time-dependent curved surfaces. In *Proceedings of ACM Siggraph*, pages 321–334, 1993.

[49] W. P. Wang and K. K.Wang. Geometric modeling for swept volume of moving solids. *IEEE Computer Graphics and Applications*, 6(12):8–17, 1986.

[50] M. Zefran and V. Kumar. Interpolation schemes for rigid body motions. *Computer-Aided Design*, 30(3):179–189, 1998.