Fast Convex 3D Polygons Overlap Testing

Philippe Guigue

Abstract

This paper presents a fast method for computing whether or not two convex polygons in three dimensions intersect. The test relies exclusively on the sign of 4×4 determinants. The source code is available online.

Keywords: geometric predicates

1 Introduction

We propose a generalization of the scheme used in the three-dimensional triangle-triangle overlap test [1] to the overlap detection of convex 3D polygons.

2 Preliminaries

Definition 2.1 *Given four three-dimensional points* $\boldsymbol{a} = (a_x, a_y, a_z)$, $\boldsymbol{b} = (b_x, b_y, b_z)$, $\boldsymbol{c} = (c_x, c_y, c_z)$, *and* $\boldsymbol{d} = (d_x, d_y, d_z)$, *we define the determinant*

$$[\boldsymbol{a}, \boldsymbol{b}, \boldsymbol{c}, \boldsymbol{d}] := - \begin{vmatrix} a_x & b_x & c_x & d_x \\ a_y & b_y & c_y & d_y \\ a_z & b_z & c_z & d_z \\ 1 & 1 & 1 & 1 \end{vmatrix} = (\boldsymbol{d} - \boldsymbol{a}) \cdot ((\boldsymbol{b} - \boldsymbol{a}) \times (\boldsymbol{c} - \boldsymbol{a})) \tag{1}$$

The sign of $[\boldsymbol{a}, \boldsymbol{b}, \boldsymbol{c}, \boldsymbol{d}]$ has two geometric interpretations, each corresponding to a right-hand rule. It tells whether vertex $\boldsymbol{d}$ is above, below, or on a plane through $\boldsymbol{a}$, $\boldsymbol{b}$, and $\boldsymbol{c}$, where *above* is the direction of a right-handed screw at $\boldsymbol{a}$ that turns from $\boldsymbol{b}$ toward $\boldsymbol{c}$. Equivalently, it tells whether a screw directed along the ray $\overrightarrow{\boldsymbol{ab}}$ turns in the direction of $\overrightarrow{\boldsymbol{cd}}$ (see Figure 1.a and 1.b). In either interpretation, the result is zero iff the four points are coplanar.

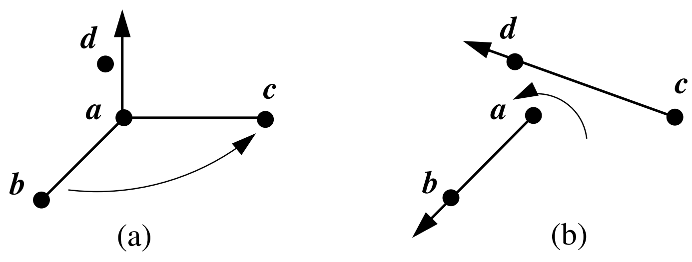

Figure 1: $[\boldsymbol{a}, \boldsymbol{b}, \boldsymbol{c}, \boldsymbol{d}]$ is positive according to a right-hand rule for planes or for lines.

In the sequel, P and Q denotes two convex three dimensional polygons with vertices $p_0, p_1, \ldots, p_n$ and $q_0, q_1, \ldots, q_m$ respectively. π_P and π_Q denote their respective supporting planes, and L the line of intersection of the two planes (see Figure 1).

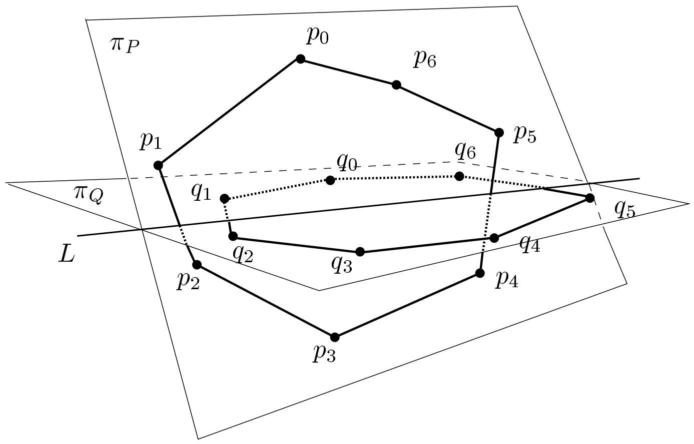

Figure 2: Polygons and the planes in which they lie.

The overlap test described in this paper returns a boolean value which is true if the closed input polygons (i.e. the polygons including their boundary) intersect. The algorithm only handles strictly convex polygons (a polygon is strictly convex if each vertex of the polygon has an interior angle strictly inferior than π.

3 Overlap Test Method

Each input polygon is a subset of the plane it lies in, so for two polygons to intersect they must intersect along the line of intersection of their planes. Hence, a necessary condition for intersection is that each polygon must intersect the plane of the other. Letting $L(t) = t.N + d$ be the line of intersection of the two planes containing P and Q, the intersection of each input polygon with the plane of the other polygon forms an interval on L. These intervals overlap if and only if the two input polygons overlap as well.

The steps of the algorithm are as follows. First, find two edges of Q (resp. of P) that straddle the plane of P (resp. of Q) in opposite directions. If such edges do not exist either at least one polygon does not intersect the plane of the other polygon or the polgons are coplanar. The first case is detected and rejected trivially as soon as at least one vertex does not lie on the other polygon plane. The latter case occurs when all vertices of one polygon lie on the other polygon plane and is processed by performing a two-dimensional polygon-polygon overlap test after projecting the vertices onto the axis aligned plane where the areas of the polygons are maximized.

Otherwise, letting $\overrightarrow{u_0u_1}$ and $\overrightarrow{u_2u_3}$ (resp. $\overrightarrow{v_0v_1}$ and $\overrightarrow{v_2v_3}$) be two edges of P (resp. of Q) that go from the positive side to the negative side of π_Q and from the negative side to the positive side of π_Q respectively (see Figure 3), the overlap test reduces to check the following conjunction

$$[u_0, u_1, v_0, v_1] \leq 0 \ \wedge \ [u_2, u_3, v_3, v_2] \leq 0$$

This can be easily seen by looking at Figure 1 and recalling the second interpretation of the three-dimensional orientation test. Indeed, this conjunction amounts to test whether a screw directed along the ray $\overrightarrow{u_0u_1}$ (resp. $\overrightarrow{u_2u_3}$) turns in the direction of $\overrightarrow{v_1v_0}$ (resp. $\overrightarrow{v_2v_3}$) (see Figure 3).

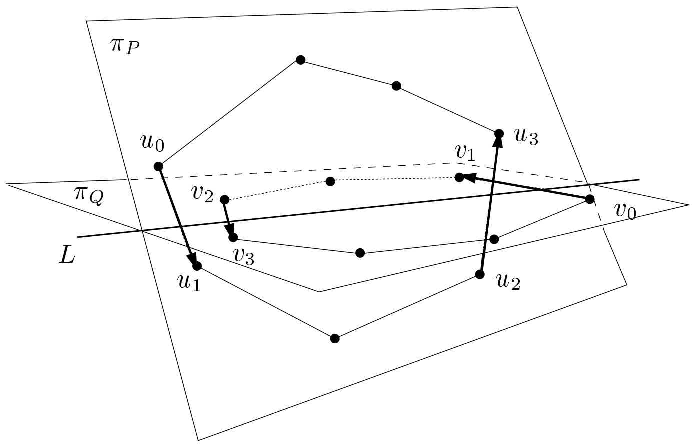

Figure 3: Once the edges u_0u_1, u_2u_3, v_0v_1 and v_2v_3 have been determined, the overlap test returns the value of $[u_0, u_1, v_0, v_1] \leq 0 \ \wedge \ [u_2, u_3, v_3, v_2] \leq 0$.

4 Implementation

Given an edge p_ip_j of P, the vector p_ip_j straddles π_Q from the positive side to the negative side if and only if $[q_0, q_1, q_2, p_i]$ and $[q_0, q_1, q_2, p_j]$ do not have the same sign and $[q_0, q_1, q_2, p_i] > 0$ or $[q_0, q_1, q_2, p_j] < 0$. Similarly, an edge p_ip_j of P straddles π_Q from the negative side to the positive side if and only if $[q_0, q_1, q_2, p_i]$ and $[q_0, q_1, q_2, p_j]$ do not have the same sign and $[q_0, q_1, q_2, p_i] < 0$ or $[q_0, q_1, q_2, p_j] > 0$.

Coplanar polygon pairs are detected when all vertices of one polygon are coplanar with the other polygon plane.

Finally, if all vertices of one polygon lies on one side of the other polygon plane (i.e. all vertices have the same sign with respect to the triangle defined by the three first other polygon's vertices), then at least one polygon does not intersect the plane of the other and there can be no intersection.

Note that the computation of the determinants can be optimized by computing the invariant parts consisting of 2×2 sub-determinants only once. For $[q_0, q_1, q_2, X]$, expansion by 2×2 sub-determinants gives the normal vector of the plane passing through q_0, q_1, and q_2, and reduces the determinants to a dot product of 3-tuples.

References

[1] Philippe Guigue, Olivier Devillers. Fast and Robust Triangle-triangle Overlap Test using Orientation Predicates. *Journal of graphics tools*, to appear.