ARTICLE IN PRESS

Computer-Aided Design ▪ (▪▪▪▪) ▪▪▪–▪▪▪

Contents lists available at ScienceDirect

Computer-Aided Design

journal homepage: www.elsevier.com/locate/cad

Detecting the intersection of two convex shapes by searching on the 2-sphere[☆]

Samuel Hornus

Inria, Villers-lès-Nancy, F-54600, France
Université de Lorraine, loria, umr 7503, Vandœuvre-lès-Nancy, F-54506, France
Cnrs, loria, umr 7503, Vandœuvre-lès-Nancy, F-54506, France

ARTICLE INFO

Keywords:
Intersection detection
GJK
Numerical robustness
Collision detection
Minkowski sum
Gauss map

ABSTRACT

We take a look at the problem of deciding whether two convex shapes intersect or not. We do so through the well known lens of Minkowski sums and with a bias towards applications in computer graphics and robotics. We describe a new technique that works explicitly on the unit sphere, interpreted as the sphere of directions. In extensive benchmarks against various well-known techniques, ours is found to be slightly more efficient, much more robust and comparatively easy to implement. In particular, our technique is compared favorably to the ubiquitous algorithm of Gilbert, Johnson and Keerthi (GJK), and its decision variant by Gilbert and Foo. We provide an in-depth geometrical understanding of the differences between GJK and our technique and conclude that our technique is probably a good drop-in replacement when one is not interested in the actual distance between two non-intersecting shapes.

1. Introduction

This paper is concerned with the problem of detecting the intersection of two convex objects. Given two convex objects A and B in $\mathbb{R}^3$, we ask whether A and B have a point in common or not. When that is the case we say that "they intersect" or "they touch each other". The intersection detection problem is a component of collision detection for more general shapes, which plays a major role in robotics [1], computer animation [2] and mechanical simulation for example.

Intersection detection also plays a role in computer graphics in general as an ingredient in acceleration data structures, such as bounding volume hierarchies or Kd-trees. In the latter case, an object of interest (*e.g.* a ray, a view frustum, or another hierarchy) is tested for intersection against the geometric shapes that bound each node in the hierarchy. These bounding shapes are simple shapes (boxes, spheres) for which fast intersection detection techniques exist. For an in-depth exposition to intersection and collision detection, we refer the reader to the book of Ericson [3] and the survey of Jiménez et al. [4].

In robotics or computer graphics, we often limit ourselves to constant-size or small convex objects, and techniques that do not use pre-processing, but the intersection detection problem has several variants and has been studied by theoreticians as well. Computational geometers have recently developed an optimal solution for general convex polyhedra: Given any collection of convex polyhedra in $\mathbb{R}^3$, one can pre-process them in linear time, independently of each other, so that the intersection of any two polyhedra P and Q from the collection can be tested in optimal time $O(\log|P| + \log|Q|)$ (see [5] and the other references within). This essentially closes the (theoretical) problem for the case of convex polyhedra. It is not clear if their technique is amenable to an efficient implementation since the data-structures used are not simple and it requires knowledge of the connectivity between the polyhedron neighboring facets.

In this paper however, we only consider techniques that do not use excessive or complex pre-processing [1] and are asymptotically slower, but very fast in practice.

Of particular interest is the beautiful algorithm developed by Gilbert, Johnson and Keerthi (GJK) for computing the distance $d(A, B)$ between two convex polyhedra A and B [6]. It only needs access to the vertices of the polyhedra and typically uses just a few iterations over them to compute the distance $d(A, B)$. It was later generalized and adapted to the decision version of the problem by Gilbert and Foo [1] to handle a broader class of convex objects. We describe these algorithms in Section 3.

In this paper we view the decision problem as that of finding an *oriented* plane that sets A on its negative and B on its positive side. Writing $\mathcal{S}^-$ for the set of normals to the planes achieving

☆ This paper has been recommended for acceptance by Mario Botsch, Yongjie Jessica Zhang and Stefanie Hahmann.

E-mail address: samuel.hornus@inria.fr.

1 Except, briefly, in Section 6.4.

http://dx.doi.org/10.1016/j.cad.2017.05.009
0010-4485/© 2017 Elsevier Ltd. All rights reserved.

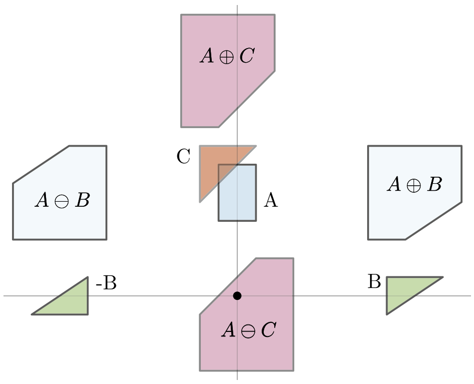

Fig. 1. Three convex shapes A, B and C lie in the Euclidean plane whose origin is marked. Some Minkowski sums and differences are drawn. Note how the origin lies in the Minkowski difference of A and C, proving that these two shapes intersect.

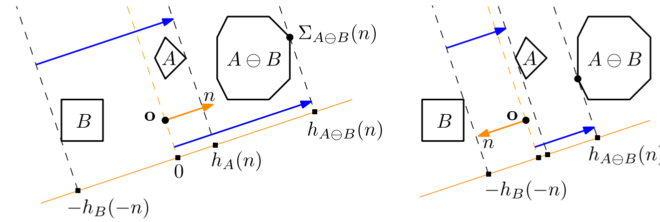

Fig. 2. Two illustrations of Eq. (8) with opposite normal vectors n. In each case, the blue vectors are identical. On the left, $\max_{x\in A\ominus B} n \cdot x = h_{A\ominus B}(n)$ is positive. On the right, it is negative, which certifies that $A \ominus B$ does not contain the origin.

separation, we design, in Section 4, an algorithm to find a direction in $\mathcal{S}^-$ or decide that $\mathcal{S}^-$ is empty (when A and B do touch each other). Our algorithm iteratively prunes parts of the unit sphere $\mathcal{S}$ so that the remaining part, a convex spherical polygon, provides an increasingly tight superset of $\mathcal{S}^-$. We call it *Decision Sphere Search*, or DSS for short.

Section 5 gives a theoretical analysis of DSS and an extensive comparison of DSS with GJK. In particular, we show that DSS optimally aggregates the information gathered about the Minkowski difference $A \ominus B$ during the successive iterations.

In Section 6, we benchmark our implementations of a "naive", quadratic algorithm, of DSS and of GJK. Each benchmark considers a specific type of objects and measures the performance of the algorithm with respect to the "collision density", *i.e.* the ratio of the number of tested pairs of convex objects that actually intersect to the total number of tested pairs. Our DSS technique appears to be faster than GJK in general, numerically more robust and easier to implement. We have taken care to analyze a large variety of situations including very uneven ones, such as frustum-culling where one object (the frustum) is much larger than the other. In that case, we show that a hybrid technique combining DSS and GJK gives the overall best results and we explain why.

2. Preliminaries

It is simpler to work with an alternative view of the geometry of the intersection detection problem, thereby reducing it to deciding whether a closed convex set contains the origin or not. The rest of the present section describes this well known alternative view.

The *Minkowski sum and Minkowski difference* of two subsets A and B of $\mathbb{R}^3$ are (see Fig. 1):

$$A \oplus B = \{a + b \mid a \in A, b \in B\} \text{ and} \tag{1}$$
$$A \ominus B = \{a - b \mid a \in A, b \in B\}. \tag{2}$$

Using this definition, A and B have non-empty intersection if and only if $A \ominus B$ contains the origin $\mathbf{o}$:

$$A \cap B \neq \emptyset \quad \Leftrightarrow \quad \mathbf{o} \in A \ominus B. \tag{3}$$

We write $d(A, B)$ for the distance between the two subsets A and B:

$$d(A, B) = \min_{a\in A, b\in B} |a - b|. \tag{4}$$

When A and B are closed convex sets, their Minkowski difference is a closed convex set as well. In Eq. (3), we have thus reduced our question on the existence of an intersection between A and B to a question regarding a single convex object $P = A \ominus B$ and the origin. In the rest of the paper we most often consider this single closed convex set P and its relation with the origin $\mathbf{o}$, instead of considering A and B separately.

We now define three functions that play an important role in the algorithms discussed in this paper. For a closed convex subset P of $\mathbb{R}^3$, we define the *support function* h_P that maps a unit vector n of the unit sphere $\mathcal{S}$ to a real number defined as

$$h_P(n) = \max_{p\in P}(n \cdot p). \tag{5}$$

The support function is closely tied to the study of convex shapes and serves as a tool to represent and manipulate them. The usefulness of the support function for shape modeling operations was recognized by Sabin who showed how offset and convolution are easily expressed with it [7]. A more technical overview with an application to the computation of Minkowski sums is given, *e.g.* by Šír et al. [8]. We also define the *extremal function* Σ_P as

$$\Sigma_P(n) = \operatorname*{argmax}_{p\in P}(n \cdot p), \tag{6}$$

i.e., $h_P(n) = n \cdot \Sigma_P(n)$. For a given direction n, several points on P might realize the largest dot-product with n. In this case we choose a point arbitrarily. When the convex set P is not bounded, we implicitly restrict the functions h_P and Σ_P to the portion of $\mathcal{S}$ where they are well defined. The *closest-point function* ν maps P to the unique point in P that realizes the distance from P to the origin, *i.e.* $\nu(P) \in P$ and $|\nu(P)| = d(P, \{\mathbf{o}\})$.

The case $P = A \ominus B$ is of particular importance for us. In this case, the functions h_P and Σ_P are computed as:

$$h_{A\ominus B}(n) = \max_{a\in A}(n \cdot a) - \min_{b\in B}(n \cdot b) \tag{7}$$
$$= h_A(n) + h_B(-n) \text{ and} \tag{8}$$
$$\Sigma_{A\ominus B}(n) = \Sigma_A(n) - \Sigma_B(-n) \text{ (see Fig. 2)}. \tag{9}$$

The convex shape $P = A \ominus B$ contains the origin if and only if its support function h_P, takes a non-negative value over the whole unit sphere:

$$A \cap B \neq \emptyset \ \Leftrightarrow\ \mathbf{o} \in P \ \Leftrightarrow\ \forall n \in \mathcal{S}, h_P(n) \geq 0, \text{ or} \tag{10}$$
$$A \cap B = \emptyset \ \Leftrightarrow\ \mathbf{o} \notin P \ \Leftrightarrow\ \exists n \in \mathcal{S}, h_P(n) < 0. \tag{11}$$

3. Related work

Research on practical intersection detection techniques for *static* convex shapes has not been very active in recent years, so we can refer the reader to the survey of Jiménez et al. [4] and the excellent book of Ericson [3]. We also refer the reader to our technical report [9], where we propose a review of previous work on intersection detection problem for small convex polyhedra with or without specific symmetries, through the unifying lens of Minkowski sums and the Gauss map, which both play a central role