Determining Directional Contact Range of Two Convex Polyhedra

Yi-King Choi[a,*], Xueqing Li[b], Fengguang Rong[b], Wenping Wang[a], Stephen Cameron[c]

[a] *The University of Hong Kong, Pokfulam Road, Hong Kong, China*
[b] *Shandong University, Shandong, China*
[c] *Oxford University Computing Laboratory, Parks Road, Oxford, OX1 3QD, U.K.*

Abstract

The *directional contact range* of two convex polyhedra is the range of positions that one of the polyhedron may locate along a given straight line so that the two polyhedra are in collision. Using the contact range, one can quickly classify the positions along a line for a polyhedron as "safe" for free of collision with another polyhedron, or "unsafe" for the otherwise. This kind of contact detection between two objects is important in CAD, computer graphics and robotics applications. In this paper we propose a robust and efficient computation scheme to determine the directional contact range of two polyhedra. We consider the problem in its dual equivalence by studying the Minkowski difference of the two polyhedra under a duality transformation. The algorithm requires the construction of only a subset of the faces of the Minkowski difference, and resolves the directional range efficiently. It also computes the contact configurations when the boundaries of the polyhedra are in contact.

Key words: directional contact range, separating distance, penetrating distance, convex polyhedra, duality transformation, signed distance

*Corresponding author

Email addresses: ykchoi@cs.hku.hk (Yi-King Choi), xqli@sdu.edu.cn (Xueqing Li), rfguang@sdu.edu.cn (Fengguang Rong), wenping@cs.hku.hk (Wenping Wang), Stephen.Cameron@comlab.ox.ac.uk (Stephen Cameron)

1. Introduction

The collision status of two objects, i.e., whether they are separate or intersecting, as well as their contact configurations, i.e., at which parts they are in contact, are important in many applications in CAD, computer graphics and robotics, or other areas that involve physical simulations, where responses are subsequently deduced based on these pieces of information. In this paper, we focus on the collision status and contact configurations of two convex polyhedra, assuming that they may only move along a given direction. The restriction regarding the direction is deemed reasonable, as there are a lot of applications in which object translations are only allowed in some specific directions. In industrial modeling or motion design, for example, the directions of movements that a mechanical part can take are limited by the constraints imposed by the degree of freedom of the part. The directional collision status of two objects is therefore useful, e.g., for object placements and motion design in a dynamic environment.

We define the *directional contact range* (DCR) of two convex polyhedra P and Q with respect to a direction $\mathbf{s}$ to be the range of positions that Q can locate along $\mathbf{s}$ so that P and Q are in contact or overlap, assuming that P is kept static. Equivalently, we say that

$$\mathrm{DCR}(P, Q, \mathbf{s}) = [\underline{\alpha}, \overline{\alpha}] \quad \text{iff} \quad P \cap Q^{t\hat{\mathbf{s}}} \neq \emptyset, \forall t \in [\underline{\alpha}, \overline{\alpha}]$$

where $\underline{\alpha}, \overline{\alpha} \in \mathbb{R}$, $\hat{\mathbf{s}} = \mathbf{s}/\|\mathbf{s}\|$ and $Q^{t\hat{\mathbf{s}}} = \{\mathbf{q} + t\hat{\mathbf{s}} \mid \mathbf{q} \in Q\}$ is the result of Q translated by $t\hat{\mathbf{s}}$. In particular, $Q^{\underline{\alpha}\hat{\mathbf{s}}}$ and $Q^{\overline{\alpha}\hat{\mathbf{s}}}$ are in external contact with P, i.e., they touch P only at some boundary points.

The DCR essentially gives the relative positions between the polyhedra at which they are in contact, and therefore can solve collision queries when Q is considered moving along $\mathbf{s}$. Since the polyhedra are convex, it is obvious that the DCR is either empty or is a single closed interval. The DCR captures the collision status of P and Q along $\mathbf{s}$, so that one can easily tell where P and Q should collide in this direction. The directional separating distance or penetration distance of P and Q, when they are separate or overlap, respectively, can also be computed from the DCR, so that if $\mathrm{DCR}(P, Q, \mathbf{s}) = [\underline{\alpha}, \overline{\alpha}]$, the required directional distance is given by $\min\{|\underline{\alpha}|, |\overline{\alpha}|\}$.

Object interference testing or collision detection has been intensively studied in the fields of computational geometry, computer graphics and robotics (see a survey in [1, 2]). Given two convex polyhedra, Cameron and Culley considered their minimum translational distance [3]; there are

feature-based algorithms [4, 5] that determine their closest features. The GJK [6] algorithm works on the simplices of the Minkowski difference of two polyhedra and uses convex optimization techniques to compute the closest points. Agarwal et al. [7] presented a randomized algorithm to determine the penetration depth of two convex polyhedra with an expected running time $\mathcal{O}(m^{\frac{3}{4}+\epsilon} + n^{\frac{3}{4}+\epsilon} + m^{1+\epsilon} + n^{1+\epsilon})$, for any constant $\epsilon > 0$, where m and n are the number of faces of the two polyhedra. Further to this theoretical result, Kim et al. [8] estimated the penetration depth of two intersecting polyhedra using the Gaussian map of their Minkowski sum.

In relation to directional contact, Dobkin et al. devised an $\mathcal{O}(\log^2 n)$ algorithm to compute the directional penetration depth of two intersecting convex polyhedra [9], and showed that the directional distance corresponds to the directional distance between the origin and the Minkowski difference polyhedron, M, of the polyhedra. Hence, a brute-force algorithm for finding the directional distance by intersecting a line from the origin and M has $\mathcal{O}(n^2)$ complexity, which is also the geometric complexity of M. There are efficient solutions for computing the intersection of line and a convex polyhedron, including linear programming approaches or geometrical methods such as [10, 11] that transform the problem to locating a point in a convex plane partition in the dual space. Our algorithm differs by using another form of duality transformation, and most importantly, we exploits the fact that M is not a general convex polyhedron, but the Minkowski difference of two convex polyhedron with much simpler geometric complexity.

1.1. Major contributions

In this paper, we present an algorithm to compute the directional contact range (DCR) of two convex polyhedra efficiently. The goal of the algorithm is to seek a face on the Minkowski difference of the two polyhedra which gives the contact features at their touching positions, given that one of them may move freely along a specific direction. We define a convex function which guarantees convergence and therefore guides the search in a robust manner. Moreover, we break down the search on the Minkowski difference into three different phases (corresponding to the three different types of faces), skipping most of the EE-type faces which is of $\mathcal{O}(n^2)$ where n is the number of faces of the polyhedra (Section 3.2), and thereby obtaining the target face efficiently.

The essence of our idea is to consider the DCR problem in its dual equivalence. We study the Minkowski difference under a duality transformation and a convex function is then defined as the signed distance of a vertex on

the dual polyhedron to a plane. We also show that maximizing the convex function is essentially the same as to finding a face containing the intersection of a ray from the origin with the Minkowski difference in the primal space, hence solving the DCR problem. The convex nature of the search process is difficult to perceive in the primal space intuitively, but could be proved easily by its dual counterpart. Although our algorithm is based on the concept of duality transformation, its computation does not involve any explicit application of the transformation and therefore no overhead is incurred in this regard.

2. The Key Idea

In this section, we explain the fundamental idea of our algorithm, which relates the DCR problem of two polyhedra in the primal space to a search for a vertex on a Minkowski difference polyhedron in the dual space. Let P be a convex polyhedron in $\mathbb{E}^3$, then $\mathcal{V}_P$, $\mathcal{F}_P$, and $\mathcal{E}_P$ denote the set of vertices, faces, and edges of P, respectively.

2.1. Minkowski difference of two polyhedra in relation to DCR

Given two polyhedra P and Q, let $-Q = \{-q \mid q \in Q\}$. We consider the Minkowski difference of P and Q (or equivalently, the Minkowski sum of P and $-Q$) defined by $M \equiv P \oplus (-Q) = \{\mathbf{p} - \mathbf{q} \mid \mathbf{p} \in P, \mathbf{q} \in Q\}$. Since P and Q are both convex, M is also a convex polyhedron [12]. The origin $\mathbf{o}$ is in M if and only if there are some $\mathbf{p} \in P$ and $\mathbf{q} \in Q$ such that $\mathbf{p} = \mathbf{q}$, i.e., P and Q overlap and share a common point. Moreover, $\mathbf{o}$ is on the boundary of M if and only if P and Q share common boundary points only.

When Q moves in a direction $\mathbf{s}$, M moves in the opposite direction $-\mathbf{s}$. If P and Q do not intersect along $\mathbf{s}$, M does not contain the origin when moving along $\mathbf{s}$ and the DCR is empty. Otherwise, the DCR is the range of distances that M can travel along $\mathbf{s}$ with the origin remains in M. In other words, the DCR are bounded by the distances from the origin to the intersections of the line $\mathbf{os}$ and the boundary of M (Fig. 1). These two intersection points corresponds to when P and Q are in external touch. In the case where $\mathbf{os}$ has only one intersection with M, the DCR is a single value which is the distance from the origin to the intersection. The intersection points must lie on the boundary of M, and hence our primary task is to compute the intersection between the line $\mathbf{os}$ and the boundary M. Since intersections

must always lie on some faces of M, faces of M (but not edges nor vertices) are only considered in our algorithm.

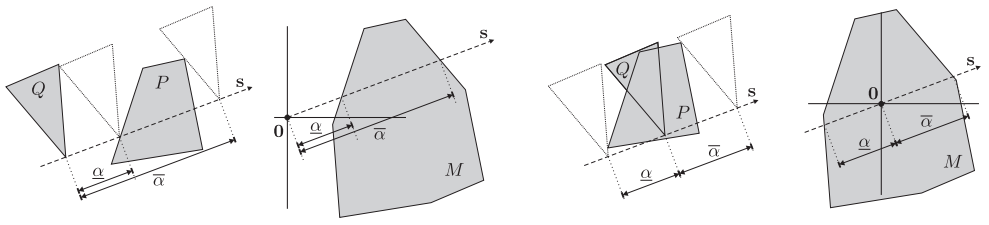


Figure 1: The DCR $[\underline{\alpha}, \overline{\alpha}]$ of two convex polyhedra P and Q, and the corresponding distances from the origin to the boundary of the Minkowski difference M when P and Q are (a) separate; and (b) overlap.

Proposition 1. *Let the line* **os** *be given by* $s(u) = u\hat{\mathbf{s}}$, $u \in \mathbb{R}$. *Suppose* **os** *intersects* $M = P \oplus -Q$ *at the faces* $f_{\min}$ *and* $f_{\max}$ *with points of intersection* $s(u_{\min})$ *and* $s(u_{\max})$, *respectively, so that* $u_{\min} \leq u_{\max}$ *and* $s(u) \in M$ *if and only if* $u \in [u_{\min}, u_{\max}]$. *Then the DCR of* P *and* Q *is given by* $[u_{\min}, u_{\max}]$.

Note that $f_{\min}$ (or $f_{\max}$) could be more than one face which happens when **os** intersects M at a vertex or an edge.

2.2. The dual of the Minkowski difference

Given an arbitrary point $\mathbf{c} \in \mathbb{R}$, we may classify the faces of M into three groups depending on the positions of $\mathbf{c}$ to the plane $\mathcal{H}_f$ containing a face f of M: f is a supporting face, a convex face or a concave face if $\mathbf{c}$ lies on $\mathcal{H}_f$, in the inner half space (i.e., the half space in which M resides) of $\mathcal{H}_f$, or in the outer half space of $\mathcal{H}_f$, respectively.

Suppose M is transformed under a duality as described in Appendix A.1 using an interior point of M as the centre of duality. Then every faces of M are properly transformed to a vertex not at infinity and the dual M^* is a convex polyhedron, with vertices $\mathcal{F}_M^*$ and faces $\mathcal{V}_M^*$. The dual of an edge defined by two adjacent vertices $\mathbf{v}_0$, $\mathbf{v}_1$ in M is an edge common to two adjacent faces $\mathbf{v}_0^*$, $\mathbf{v}_1^*$ in M^*. In general, if an arbitrary point $\mathbf{c}$ not in the interior of M is used as the centre of duality, the above correspondence between the features of M and M^* still applies, but M^* is no longer compact in $\mathbb{E}^3$. Its boundary is defined by two disjoint shells that extends to infinity. In particular, the

supporting faces (if any) of M with respect to $\mathbf{c}$ are transformed to points at infinity, while the dual of the convex faces are the convex faces w.r.t. the origin in the dual space which form one of the continuous convex shell of M^* (Fig. 2). As we shall see, our algorithm will work on the convex faces with respect to a given point.

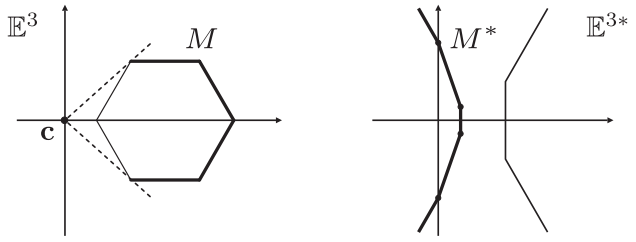


Figure 2: A 2D illustration of M and its dual M^* with the centre of duality $\mathbf{c}$ not in M. The convex faces w.r.t. $\mathbf{c}$ on M (thick lines) corresponds to the black points lying on the convex shell (marked with thick lines) with respect to the origin in $\mathbb{E}^{3*}$.

2.3. Signed distance of a face from the origin

Suppose that a plane is given by $\Pi : A^T\mathbf{x} = k$ where $A \in \mathbb{R}^3$, $\mathbf{x} = (x, y, z)^T$, $k \in \mathbb{R}$, and we assume that Π is normalized such that $\|A\| = 1$ and $k > 0$ is the shortest distance from the origin to Π. The *signed distance* of a point $\mathbf{x}_0$ to the plane Π is then given by $\mathrm{d}_\Pi(\mathbf{x}_0) = \mathbf{A}^T\mathbf{x}_0 - k$. Given a point $\mathbf{c} \notin \mathbf{M}$, let $\hat{\mathcal{F}}_\mathbf{c}$ denote the set of convex faces of M with respect to $\mathbf{c}$. We define the *signed distance* of $f \in \hat{\mathcal{F}}_\mathbf{c}$, denoted by $\mathrm{d}(f)$, to be the signed distance of f^* to the plane $\mathbf{o}^*$ in $\mathbb{E}^{3*}$, i.e., $\mathrm{d}(f) = \mathrm{d}_{\mathbf{o}^*}(f^*)$, where $\mathbf{c}$ is the centre of duality and $\mathbf{o}^*$ is the dual of the origin $\mathbf{o} \in \mathbb{E}^\mathbf{3}$. Let $\mathcal{H}_f : \mathbf{N}^T\mathbf{x} = k$, where $\|\mathbf{N}\| = 1$ and $k > 0$, be the containing plane of f. Then $f^* = \mathbf{N}/(k - \mathbf{N}^T\mathbf{c}) \in \mathcal{V}_{M^*}$ in the dual space with $\mathbf{c}$ as the centre of duality. The origin $\mathbf{o}$ is first translated by $-\mathbf{c}$ and the plane equation of $\mathbf{o}^*$ is $-\mathbf{c}^T\mathbf{x} = 1$. Hence, we have

$$\mathrm{d}(f) = \mathrm{d}_{\mathbf{o}^*}(f^*) = \frac{-\mathbf{c}^T}{\|\mathbf{c}\|} \cdot \frac{\mathbf{N}}{k - \mathbf{N}^T\mathbf{c}} - \frac{1}{\|\mathbf{c}\|} = -\frac{k}{\|\mathbf{c}\|(k - \mathbf{N}^T\mathbf{c})}. \tag{1}$$

The quantity $\mathrm{d}_{\mathbf{o}^*}(f^*)$ uniquely determines a plane $\mathbf{l}^*$ in $\mathbb{E}^{3*}$ passing through f^* and parallel to $\mathbf{o}^*$ such that $\mathrm{d}_{\mathbf{o}^*}(\mathbf{x}) = \mathrm{d}_{\mathbf{o}^*}(f^*)$ for all points $\mathbf{x} \in \mathbf{l}^*$ (Fig. 3). Since $\mathbf{l}^*$ and $\mathbf{o}^*$ have the same normal direction, it can also be shown that $\mathbf{l}$, $\mathbf{c}$ and $\mathbf{o}$ are collinear. Moreover, $\mathbf{l}^*$ passes through f^* and hence $\mathbf{l}$ must lie

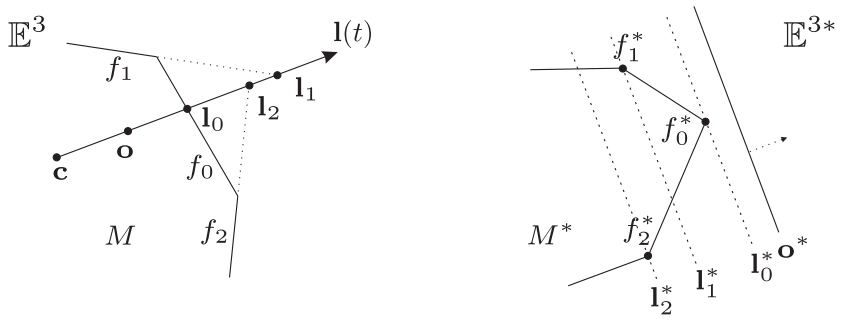


Figure 3: The vertex f_0^* in $\mathbb{E}^{3*}$ attaining maximum signed distance to $\mathbf{o}^*$ is the dual of a face f_0 in $\mathbb{E}^3$ intersecting the ray $\mathbf{co}$.

on $\mathcal{H}_f$, the containing plane of f. This implies that $\mathbf{l}$ is the intersection of $\mathcal{H}_f$ and the line $\mathbf{co}$.

Suppose that the ray $\mathbf{co}$, given by $\mathbf{l}(t) = -t\mathbf{c}, t > 0$, hits some faces in $\hat{\mathcal{F}}_\mathbf{c}$ (Fig. 3). Then the containing planes of all faces in $\hat{\mathcal{F}}_\mathbf{c}$ must intersect $\mathbf{co}$. The signed distance for a convex face $f_t \in \hat{\mathcal{F}}_\mathbf{c}$ whose containing plane intersects $\mathbf{co}$ at a point $\mathbf{l}(t)$ is $\mathrm{d}(f_t) = (1-t)/(t\,\|c\|)$. Since $\mathrm{d}(f_t)$ is a decreasing function for $t > 0$, it means that the containing plane of the face with maximum signed distance over all faces in $\hat{\mathcal{F}}_\mathbf{c}$, has the closest intersection to $\mathbf{c}$ with the ray $\mathbf{co}$. Due to the convexity of M, this intersection must lie on a face of M and we have the following proposition:

Proposition 2. *Let $f_{\max} \in \hat{\mathcal{F}}_\mathbf{c}$ be the convex face with respect to $\mathbf{c}$ whose signed distance is the maximum over all faces in $\hat{\mathcal{F}}_\mathbf{c}$, i.e., $f_{\max} = \arg\max_f\{\mathrm{d}(f) \mid f \in \hat{\mathcal{F}}_\mathbf{c}\}$. Then $f_{\max}$ contains an intersection of the ray $\mathbf{co}$ and M.*

Since f^* lies on a convex shell, the signed distance function is convex over $\hat{\mathcal{F}}_\mathbf{c}$. Starting from a face $f \in \hat{\mathcal{F}}_\mathbf{c}$, we may therefore search for $f_{\max}$ at which the ray $\mathbf{co}$ intersects M. It is important to note that the intersection needs not be solved, as its distance from the origin can be computed directly from $\mathrm{d}(f_{\max})$ as follows. We established that $\mathrm{d}(f) = (1-t)/(t\|\mathbf{c}\|)$ is the signed distance of a convex face $f \in \hat{\mathcal{F}}_\mathbf{c}$ whose containing plane intersects $\mathbf{co}$ at $\mathbf{l}(t)$. Let $\alpha(f)$ be the signed distance of $\mathbf{l}(t)$ from $\mathbf{o}$ along the ray $\mathbf{co}$. Then,

$$\alpha(f) = (t-1)\|\mathbf{c}\| = \Big(\frac{1}{\mathrm{d}(f)\|\mathbf{c}\| + 1} - 1\Big)\|\mathbf{c}\| = -\frac{\mathrm{d}(f)\|\mathbf{c}\|^2}{\mathrm{d}(f)\|\mathbf{c}\| + 1}.$$

Hence, the distance from the origin to the intersection of $f_{\max}$ and $\mathbf{co}$ is given by $\alpha(f_{\max})$.

3. The Algorithm

Given two convex polyhedra P and Q, and a direction $\mathbf{s} \in \mathbb{R}^3$, the following algorithm computes the DCR of P and Q with respect to $\mathbf{s}$:

Step 1: Check whether the line $\mathbf{os}$ intersects $M = P \oplus (-Q)$. If not, we have $\mathrm{DCR}(P, Q, \mathbf{s}) = \emptyset$. Otherwise, choose a point $\mathbf{c}_{\min} = u\mathbf{s}$ for some $u > 0$, and that both $\mathbf{o}$ and M lie on the same side of $\mathbf{c}_{\min}$ on $\mathbf{os}$. Choose also $\mathbf{c}_{\max} = v\mathbf{s}$ for some $v < 0$, with both $\mathbf{o}$ and M lying on the same side of $\mathbf{c}_{\max}$ on $\mathbf{os}$.

Step 2: Using $\mathbf{c}_{\min}$ as the centre of duality, search for $f_{\min}$ which attains the maximum signed distance among all convex faces with respect to $\mathbf{c}_{\min}$, i.e., $f_{\min} = \arg\max_f \{\mathrm{d}(f) \mid f \in \hat{\mathcal{F}}_{\mathbf{c}_{\min}}\}$. Then, use $\mathbf{c}_{\max}$ as the centre of duality and search for $f_{\max} = \arg\max_f \{\mathrm{d}(f) \mid f \in \hat{\mathcal{F}}_{\mathbf{c}_{\max}}\}$.

Step 3: Report $\mathrm{DCR}(P, Q, \mathbf{s}) = [\underline{\alpha}, \overline{\alpha}]$ where $\underline{\alpha} = -\alpha(f_{\min})$ and $\overline{\alpha} = \alpha(f_{\max})$.

Our algorithm does not require the complete construction of the Minkowski difference M. Moreover, we devise a novel search scheme in step 2 which skips some faces in M in order to reach $f_{\min}$ and $f_{\max}$ efficiently. The details would be discussed in subsequent sections.

3.1. Determining the center of duality $\mathbf{c}$

If two polyhedra P and Q do not meet no matter how far Q moves along a given direction $\mathbf{s}$, their DCR with respect to $\mathbf{s}$ is empty. In this case, the line $\mathbf{os}$ does not intersect the Minkowski difference M, which can be checked without constructing M as follows.

Let $\dot{P}$ and $\dot{Q}$ be the orthographic projection along $\mathbf{s}$ of P and Q to a plane $\mathcal{H}$ normal to $\mathbf{s}$. We construct the convex hull, $CH(\dot{P})$ and $CH(\dot{Q})$, of $\dot{\mathcal{P}}$ and $\dot{\mathcal{Q}}$, respectively. This can be done efficiently since the vertices of $CH(\dot{P})$ and $CH(\dot{Q})$ are the silhouette vertices of P and Q as viewed along $\mathbf{s}$. Then, we obtain $\dot{M} = CH(\dot{P}) \oplus \big(-CH(\dot{Q})\big)$. Now, $\mathbf{os}$ intersects M if and only if $CH(\dot{P})$ and $CH(\dot{Q})$ overlap, i.e., $\dot{M}$ contains the origin.

Suppose now that $\mathbf{os}$ intersects M. In general $\mathbf{os}$ has two intersections with M which is convex, and therefore we need to choose two points, each as the centre of duality to locate one intersection at one time. To locate the face $f_{\max}$ of M (see Proposition 1), the centre of duality $\mathbf{c}_{\max}$ should lie

on **os** so that the ray $\mathbf{c}_{\max}\mathbf{o}$ hits $f_{\max}$ and that $f_{\max}$ is a convex face with respect to $\mathbf{c}_{\max}$. Hence, we require that $\mathbf{c}_{\max} = v\mathbf{s}$ for some $v < 0$, and $\mathbf{o}$ and M be on the same side of $\mathbf{c}_{\max}$ on **os**. Now, $\mathbf{c}_{\max}$ can be computed easily by approximating M with its bounding box $M_{BB} = P_{BB} \oplus -Q_{BB}$, where P_{BB} and Q_{BB} are the bounding boxes of P and Q, respectively. The point $\mathbf{c}_{\min} = u\mathbf{s}$ for some $u > 0$ is then chosen similarly.

3.2. Searching the face with maximum signed distance

Step 2 of our algorithm involves searching the faces $f_{\max}$ and $f_{\min}$ at which **os** intersects M. We will only describe the search for $f_{\max}$, since $f_{\min}$ can be found in the same way using $\mathbf{c}_{\min}$ instead as the centre of duality.

A brute-force search for $f_{\max}$ is to first construct M, which is of $\mathcal{O}(n^2)$ complexity. Moreover, to locate $f_{\max}$ directly on M using its face adjacency information is inefficient, as face traversal can only advance to an immediate neighbour at one step. We therefore break down the search for $f_{\max}$ in three successive phases, each within an independent face subset of M. This allows a quicker leap over the faces on M and hence a more rapid search of $f_{\max}$. Also, the number of faces on M that needs to be constructed are greatly reduced.

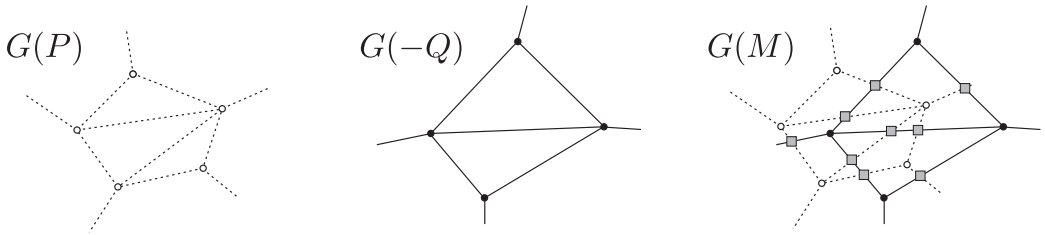


Figure 4: The planar representation of the Gaussian image $G(M)$ by superimposing $G(P)$ and $G(-Q)$. There are three types of vertices in $G(M)$: (i) (white point) a point of $G(P)$ falling within a region of $G(-Q)$, i.e., a face in $\mathcal{F}_{\text{fv}}$; (ii) (black point) a point of $G(-Q)$ falling within a region of $G(P)$, i.e., a face in $\mathcal{F}_{\text{vf}}$; and (iii) (shaded square) the intersection point of two arcs, each from $G(P)$ and $G(-Q)$, i.e., a face in $\mathcal{F}_{\text{ee}}$.

Let us define the supporting vertex of of a polyhedron P for a face f be $\mathrm{s}_P(f) = \arg\max_{\mathbf{v}}\{\mathbf{n}(f) \cdot \mathbf{v} \mid \mathbf{v} \in \mathcal{V}_P\}$, where $\mathbf{n}(f)$ is the normal vector of f and $\cdot$ denote the vector dot-product. The Gaussian image of M, $G(M)$, is obtained by superimposing the Gaussian images $G(P)$ and $G(-Q)$ (Appendix A.2). For any face $f_p \in \mathcal{F}_P$, the point $G(f_p)$ must fall within the region $G\big(\mathrm{s}_{-Q}(f_p)\big)$. Similarly, the point $G(f_q)$ must fall within the region

$G\big(\mathrm{s}_P(f_q)\big)$. Hence, each point in $G(P)$ and $G(-Q)$ corresponds to a face of FV- and VF-type, respectively, in M (Fig. 4). Furthermore, each arc-arc intersection on $\mathcal{S}^2$ corresponds to a pair of edges (one from P and one from $-Q$) sharing a common normal direction and amounts to a EE-type face in M. The FV-, VF- and EE-type faces form three independent subsets $\mathcal{F}_{\mathrm{fv}}$, $\mathcal{F}_{\mathrm{vf}}$ and $\mathcal{F}_{\mathrm{ee}}$, respectively, which are given as follows (Fig. 5):

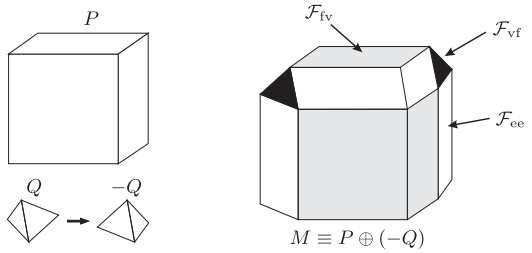


Figure 5: The Minkowski difference M of P and Q, and its three sets of faces $\mathcal{F}_{\mathrm{fv}}$, $\mathcal{F}_{\mathrm{vf}}$, and $\mathcal{F}_{\mathrm{ee}}$.

$\mathcal{F}_{\mathrm{fv}}$: Each face $F(f_p, \mathbf{v}_q)$ is a point set $\{\mathbf{x} + \mathbf{v}_q \mid \mathbf{x} \in f_p\}$, where $f_p \in \mathcal{F}_P$ and $\mathbf{v}_q \in \mathcal{V}_{-Q}$. Also, $\mathbf{v}_q = \mathrm{s}_{-Q}(f_p)$.

$\mathcal{F}_{\mathrm{vf}}$: Each face $F(\mathbf{v}_p, f_q)$ is a point set $\{\mathbf{v}_p + \mathbf{x} \mid \mathbf{x} \in f_q\}$, where $f_q \in \mathcal{F}_{-Q}$ and $\mathbf{v}_p \in \mathcal{V}_P$. Also, $\mathbf{v}_p = \mathrm{s}_P(f_q)$.

$\mathcal{F}_{\mathrm{ee}}$: Each face $F(e_p, e_q)$ is a parallelogram with vertices $\mathbf{v}_0 = \mathbf{v}_{p_0} + \mathbf{v}_{q_0}, \mathbf{v}_1 = \mathbf{v}_{p_1} + \mathbf{v}_{q_0}, \mathbf{v}_2 = \mathbf{v}_{p_1} + \mathbf{v}_{q_1}, \mathbf{v}_3 = \mathbf{v}_{p_0} + \mathbf{v}_{q_1}$ where $\mathbf{v}_{p_0}, \mathbf{v}_{p_1} \in \mathcal{V}_P$, $\mathbf{v}_{q_0}, \mathbf{v}_{q_1} \in \mathcal{V}_{-Q}$, and $e_p = (\mathbf{v}_{p_0}, \mathbf{v}_{p_1}) \in \mathcal{E}_P$, $e_q = (\mathbf{v}_{q_0}, \mathbf{v}_{q_1}) \in \mathcal{E}_{-Q}$. Moreover, the Gaussian images of e_p and e_q intersect on $\mathcal{S}^2$ (Fig. 4).

The following pseudocode searches for $f_{\max}$ with the maximum signed distance $d_{\max}$ among all convex faces with respect to $\mathbf{c}$ on $M = P \oplus -Q$:

Procedure MaxSignedDistance(P, Q, $\mathbf{c}$)
  $(f_{\mathrm{fv}}, d_{\mathrm{fv}}) \leftarrow$ Search-FV
  $(f_{\mathrm{vf}}, d_{\mathrm{vf}}) \leftarrow$ Search-VF
  $(f_{\max}, d_{\max}) \leftarrow$ Search-EE($d_{\mathrm{fv}}, d_{\mathrm{vf}}$)
  return $(f_{\max}, d_{\max})$

3.2.1. Search-FV

This procedure is to search for a face with the maximum signed distance among all convex faces with respect to $\mathbf{c}$ in $\mathcal{F}_{\text{fv}}$. The search is conducted according to face adjacency of P. It is important that we start from a convex face on M, which ensures that all subsequent faces in the search are convex faces, due to the convexity of the signed distance function. We choose $f_0 = \arg\max_f\{\mathbf{n}(f)\cdot\mathbf{co} \mid f \in \mathcal{F}_P\}$, where $\mathbf{n}(f)$ is the normal vector of a face f, as the initial face such that the corresponding face $F\big(f_0, \mathrm{s}_{-Q}(f_0)\big) \in \mathcal{F}_{\text{fv}}$ is guaranteed to be a convex face with respect to $\mathbf{c}$. Starting from f_0, the search in Search-FV considers the neighbouring faces of the current face in P and advances to one whose corresponding face in M has the local maximum signed distance. Neighbouring (or adjacent) faces are those faces incident to the vertices of the current face in P. Two faces adjacent in P may not constitute adjacent faces in M, and therefore a gain (by skipping some faces in M) is obtained by advancing faces in the search based on their adjacency in P.

The procedure is described in the following pseudocode. The function SignedDistance-FV(f) constructs a face $F\big(f, \mathrm{s}_{-Q}(f)\big) \in \mathcal{F}_{\text{fv}}$ and computes its signed distance using Eq. (1). The supporting vertex of $-Q$ for a face f is determined using the hierarchical representation of a polyhedron by Dobkin and Kirkpatrick [13].

Procedure Search-FV
  d_{fv} = SignedDistance-FV(f_0)
  For each iteration i,
    For each of the n faces f_i^j, $j = 0, \ldots, n-1$, that are adjacent to f_i in P,
      $d_i^j \leftarrow$ SignedDistance-FV(f_i^j).
    If $d_{\text{fv}} < d_i^k$, where $d_i^k = \max\{d_i^j\}$,
      $d_{\text{fv}} \leftarrow d_i^k$, $f_{i+1} \leftarrow f_i^k$.
    Else,
      Return (f_i, d_{fv}).

Theorem 1 states the correctness of Search-FV whose proof can be found in Appendix B.

Theorem 1 (Correctness of Search-FV)**.** *The face f_{fv} returned by Search-FV attains the maximum signed distance d_{fv} among all convex faces in $\mathcal{F}_{\text{fv}}$ with respect to* $\mathbf{c}$*, i.e., $f_{\text{fv}} = \arg\max_f \{\mathrm{d}(f) \mid f \in \mathcal{F}_{\text{fv}} \cap \hat{\mathcal{F}}_{\mathbf{c}}\}$.*

3.2.2. Search-VF

This procedure searches for a face with maximum signed distance among all convex faces with respect to $\mathbf{c}$ in $\mathcal{F}_{\text{vf}}$. The face $f_{\text{fv}} = F\big(f_p, \mathrm{s}_{-Q}(f_p)\big) \in \mathcal{F}_{\text{fv}}$ computed by Search-FV is supposed to be closest to $f_{\max}$ among all faces in $\mathcal{F}_{\text{fv}}$, and it should give a good starting point for subsequent search. Hence, we choose the initial face for Search-VF as a face f_0 that is incident at $\mathrm{s}_{-Q}(f_p)$ in $-Q$. The search then proceeds like Search-FV by interchanging the role of P and $-Q$. Similarly, we have the following theorem.

Theorem 2 (Correctness of Search-VF)**.** *The face f_{vf} returned by Search-VF attains the maximum signed distance d_{vf} among all convex faces in $\mathcal{F}_{\text{vf}}$ with respect to* $\mathbf{c}$*, i.e., $f_{\text{vf}} = \arg\max_f \{\mathrm{d}(f) \mid f \in \mathcal{F}_{\text{vf}} \cap \hat{\mathcal{F}}_{\mathbf{c}}\}$.*

3.2.3. Search-EE

The two procedures Search-FV and Search-VF determine the faces f_{fv} and f_{vf} with the maximum signed distance d_{fv} and d_{vf} among all convex faces with respect to $\mathbf{c}$ in the set $\mathcal{F}_{\text{fv}}$ and $\mathcal{F}_{\text{vf}}$, respectively. The next step is to search for the remaining convex faces in $\mathcal{F}_{\text{ee}}$, starting from f_{fv} or f_{vf}, whichever attains the greater signed distance. Let e_p and e_q be edges in $\mathcal{E}_P$ and $\mathcal{E}_{-Q}$, respectively. A face $F(e_p, e_q) \in \mathcal{F}_{\text{ee}}$ is formed only if the Gaussian images of e_p and e_q intersect on S^2 (Section A.2). The steps of Search-EE are given in the following pseudocode:

Procedure Search-EE

$\mathrm{d}_{\mathrm{ee}} \leftarrow \max\{d_{\mathrm{fv}}, d_{\mathrm{vf}}\}$.
$f_m \leftarrow$ the face f_{fv} or f_{vf} attains d_{ee}.
If $f_m = f_{\mathrm{fv}} = F\big(f_p, \mathrm{s}_{-Q}(f_p)\big)$, then
  $\mathcal{FS}_0 \leftarrow$ all possible faces $F(e_p, e_q)$, where e_p is an edge incident to a vertex of f_p, and e_q is an edge incident with $\mathrm{s}_{-Q}(f_p)$,
Else if $f_m = f_{\mathrm{vf}} = F\big(\mathrm{s}_P(f_q), f_q\big)$, then
  $\mathcal{FS}_0 \leftarrow$ all possible faces $F(e_p, e_q)$, where e_p is an edge incident with $\mathrm{s}_P(f_q)$, and e_q is an edge incident to a vertex of f_q.
For each iteration $i = 0, 1, 2, \ldots$
  Let $\hat{f}_i = F(\hat{e}_p, \hat{e}_q)$ be the face in $\mathcal{FS}_i$ with the maximum signed distance.
  If $d_{\mathrm{ee}} < \mathrm{d}(\hat{f}_i)$, then
    $d_{\mathrm{ee}} \leftarrow \mathrm{d}(\hat{f}_i), \quad f_{\mathrm{ee}} \leftarrow \hat{f}_i$
    $\mathcal{FS}_{i+1} \leftarrow$ all possible faces $F(e_p, e_q)$, where e_p is an edge incident to an end vertex of $\hat{e}_p$, e_q is an edge incident to an end vertex of $\hat{e}_q$
  Else
    Return $(f_{\mathrm{ee}}, d_{\mathrm{ee}})$.

We show in the Appendix that the initial face set $\mathcal{FS}_0$ contains all neighbouring EE-type faces of the initial face f_m, by considering all possible EE-type faces formed by an edge incident to the vertex that forms f_m and an edge incident with a face that forms f_m (Lemma 3). Moreover, the subsequent face sets $\mathcal{FS}_i$ includes all the neighbouring EE-type faces of the current EE-type face $\hat{f}_i$ with the maximum signed distance, by considering all possible EE-type faces formed by two edges, each incident to an end vertex of an edge forming $\hat{f}_i$ (Lemma 4). Hence, we have the following theorem:

Theorem 3 (Correctness of SEARCH-EE)**.** *The face $f_{\max}$ returned by Search-EE attains the maximum signed distance among all convex faces in $\hat{\mathcal{F}}_{\mathbf{c}}$ with respect to $\mathbf{c}$, i.e., $f_{\max} = \arg\max_f\{\mathrm{d}(f) \mid f \in \hat{\mathcal{F}}_{\mathbf{c}}\}$.*

3.3. Computation details

3.3.1. Contact configurations

The faces $f_{\min}$ and $f_{\max}$ indicate the contact configuration of P and Q when they are in external contact along the DCR direction. The contact

features are given by the features on P and Q that form the faces $f_{\min}$ and $f_{\max}$. For example, if $f_{\max} = F(\mathbf{v}_p, f_q) \in \mathcal{F}_{\text{vf}}$, the contact features of P and $Q^{\overline{\alpha}\mathbf{s}}$ are the vertex $\mathbf{v}_p \in P$ and the face $f_q \in Q$.

3.3.2. Supporting faces with respect to the centre of duality

We may encounter supporting faces with respect to the centre of duality in our algorithm, which are possible neighbours of a convex face. Supporting faces correspond to points at infinity in the dual space (Section 2.2) and can be identified if $k - N^T\mathbf{c} = 0$ when evaluating the signed distance given by Eq. (1). Supporting faces are ignored in our algorithm without affecting its correctness.

3.3.3. To decide whether two arcs on S^2 intersect.

In Search-EE, to decide whether two edges $e_p \in \mathcal{E}_P$ and $e_q \in \mathcal{E}_{-Q}$ form a face $F(e_p, e_q) \in \mathcal{F}_{\text{ee}}$, we check whether $G(e_p)$ and $G(e_q)$ intersect on the Gaussian sphere $\mathcal{S}^2$. Let $\mathbf{a}, \mathbf{b}$ be the end points of $G(e_p)$, $\mathbf{c}, \mathbf{d}$ be the end points of $G(e_q)$ and $\mathbf{o}$ be the centre of $\mathcal{S}^2$ (Fig. 6). The arcs $G(e_p)$ and $G(e_q)$ intersect if and only if (1) $\mathbf{c}, \mathbf{d}$ are on different sides of the plane $\mathbf{oba}$; (2) $\mathbf{a}, \mathbf{b}$ are on different sides of the plane $\mathbf{ocd}$; and (3) $\mathbf{a}, \mathbf{b}, \mathbf{c}, \mathbf{d}$ are on the same hemisphere. Consider the *signed volume*, $|\mathbf{cba}| = \det[\ \mathbf{c}\quad \mathbf{b}\quad \mathbf{a}\ ]$, of a parallelepiped spanned by three vectors $\mathbf{a}, \mathbf{b}, \mathbf{c}$. The above three conditions can be formulated as (1) $|\mathbf{cba}| \times |\mathbf{dba}| < 0$; (2) $|\mathbf{adc}| \times |\mathbf{bdc}| < 0$; and (3) $|\mathbf{acb}| \times |\mathbf{dcb}| > 0$. We need to compute $|\mathbf{cba}|, |\mathbf{dba}|, |\mathbf{adc}|$ and $|\mathbf{bdc}|$ only, since $|\mathbf{acb}| = |\mathbf{cba}|$ and $|\mathbf{dcb}| = |\mathbf{bdc}|$.

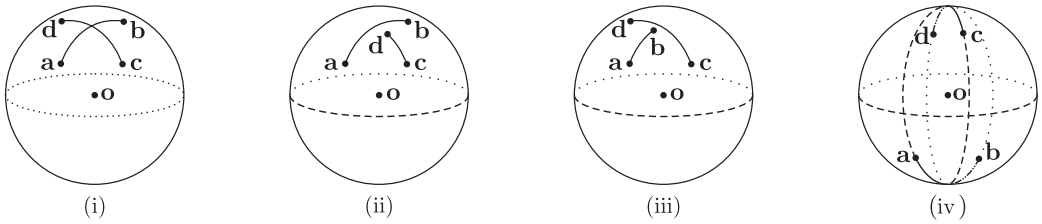


Figure 6: Determining if two arcs intersect on $\mathcal{S}^2$. Arcs intersect in (i). No intersection where (ii) condition (1); (iii) condition (2) and (iv) condition (3) is violated.

3.3.4. Span of faces with same normal direction

To simplify the preceding discussions, we assumed that all faces on M have distinct normal directions. However, this is not always true for convex

polyhedra with arbitrary mesh structures. In this case, a face is augmented to include also its neighbouring span of faces with the same normal direction.

3.3.5. To avoid repetitive visits to a face

A hash table is used to record the visited faces in each procedure to avoid unnecessary repetitive computations for a face which is visited previously in the searching process.

4. Performance

We do not compare against other existing related algorithms for computing the distance between polyhedral objects, as they are generally targetted for purposes other than computing the directional distances; hence, a direct comparison is deemed inappropriate. However, several sets of experiments are designed to evaluate the performance of our method. A set of six convex polyhedra (Fig. 7) are used (whose names and number of vertices are given in the brackets): a truncated elliptic cone (P_1 – 20), a truncated elliptic cylinder (P_2 – 50), two ellipsoids (P_3 – 200, P_4 – 500), the convex hull of a random point set in a cube (P_5 – 100), and the volume of revolution of a convex profile curve (P_6 – 200). The sizes of the polyhedra are all within a sphere of radius 5. The cone and the cylinder are with the aspect $a : b : h = 1 : 2 : 4$, where a, b are the sizes of the base ellipse and h is the height. The size of the ellipsoids are in $a : b : c = 2 : 2 : 5$ and $2 : 4 : 5$ respectively for P_3 and P_4, where a, b and c are the length of the three major axes.

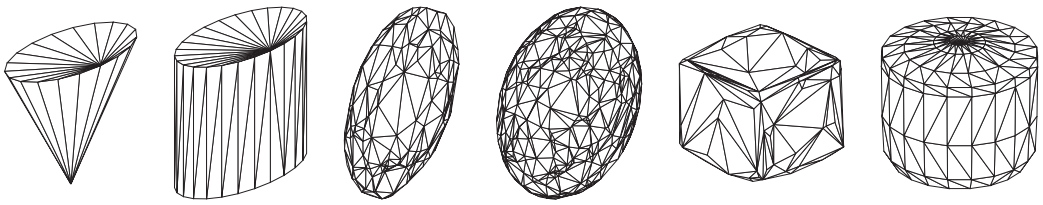


Figure 7: The six objects used in the experiments.

Five pairs of objects are chosen for DCR calculations: (P_2, P_2), (P_1, P_3), (P_6, P_6), (P_4, P_5) and (P_4, P_6), where the total number of vertices of the two objects are 100, 220, 400, 600, 700, respectively. We also generated another 10 pairs of objects which are ellipsoids of the same size aspect as P_4, but with different number of vertices. For each pair of objects P and Q, P

is kept static and Q assumes 40 random orientations; for each orientation, Q is also translated so that the shortest distance between the two objects ranges from -1.5 to 1.5 in 11 samples, of which 5 samples correspond to the cases where P and Q intersect, 1 sample corresponds to touching, and 5 samples correspond to separation. Also, for each fixed shortest distance with a random orientation, we compute the DCR between P and Q with respect to 40 random directions. It means that, for each pair of convex polyhedra, we perform a total of $11 \times 40 \times 40$ different DCR computations and the average CPU time for each run is taken. Each reported non-empty DCR $[\underline{\alpha}, \overline{\alpha}]$ along a specified direction $\mathbf{s}$ is verified by translating Q along $\mathbf{s}$ by $\underline{\alpha}$ and $\overline{\alpha}$, and using the GJK algorithm to compute the shortest distance between P and the translated Q, which should be zero as the objects are then in external contact. We note that the average shortest distance is 1.9×10^{-6} with a standard deviation of 10^{-5}; the maximum of the absolute shortest distance is found to be 10^{-4}.

The experiments were carried out on a desktop computer with an Intel Core 2 Duo E6600 2.40 GHz CPU (single-threaded) and a 2GB main memory. The performance of our algorithm is shown in Fig. 8. It takes less than 0.25 milliseconds to compute the DCR of two convex polyhedra with a total of 1000 vertices. Although the cylinder pair (P_2, P_2) is of only 100 vertices in total, the running time in this case is disproportionally longer than expected. Not only that a face at the planar bases of cylinders needs to be augmented to include the span of all other coplanar faces (Section 3.3.4), the increase in running time is also due largely to the fact that a face at the base has a large number of adjacent faces—which are the 50 faces on the curved surface of a cylinder.

The result of ellipsoid pairs shows empirically an approximately linear growth in the running time with respect to the total number of vertices. Recall that not all faces on the Minkowski difference M are being constructed and visited. From the above experiments, it is found that in a search of a single intersection on M, on average 13.7% of the faces on M is visited. In particular, only 2.5% of the EE-type faces is visited on average, which means that most of the EE-type faces are skipped in our algorithm.

A worst case scenario is designed where the number of EE-type faces is of $\mathcal{O}(n^2)$, where n is the number of vertices on the polyhedra. Two cones are constructed, each having 21 vertices (20 on the circular rim and 1 at the apex), 38 faces (20 on the slanted surface) and 57 edges. Both cones are very flat with aspect ratio $a : b : h = 12 : 12 : 1$ (Fig. 9(a)). The

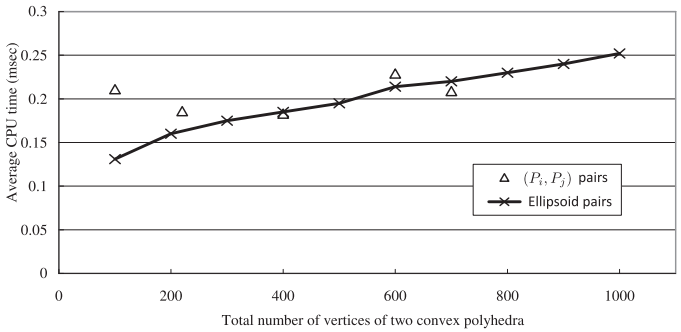


Figure 8: The average CPU time for computing the DCR of five pairs (P_i, P_j) of specific convex polyhedra $(P_1 - P_6)$, and 10 pairs of ellipsoids with varying number of vertices.

Gaussian image of each cone has one point (corresponding to the 18 faces on the flat surface) that is antipodal to a set of 20 points (corresponding to the 10 faces on the slanted surface), and 20 great arcs that looks like great semicircles, which corresponds to the edges on the circular rim (Fig. 9(b)). One of the cones is rotated about the y-axis by 90 degrees, so that each of its 20 great semicircles in the Gaussian image intersects with half of the great semicircles in the Gaussian image of the other cone (Fig. 9(b)). Hence, there are in total 38 FV-type, 38 VF-type and 200 EE-type faces on the Minkowski difference of the cones. Our algorithm, on computing a single intersection on the Minkowski difference, requires a visit to only 21 FV-type, 21 VF-type and 20 EE-type faces, showing that most of the EE-type faces (90%) are skipped which renders an efficient computation.

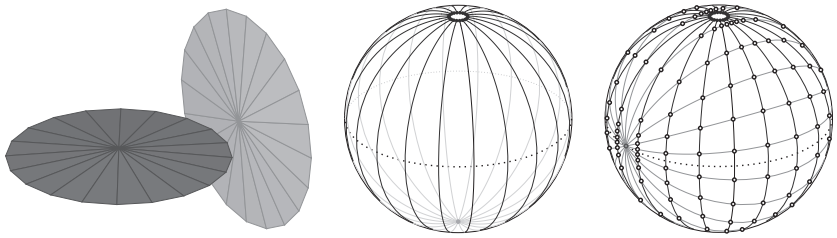


Figure 9: (a) Two thin cones whose DCR is computed. (b) Gaussian image of one of the cones. (c) Gaussian image of the Minkowski difference of the cones; the white circles correspond to the EE-type faces (only features on the front-facing surface of the Gaussian sphere are shown.)

Next, we compute the DCR of two circular cylinders, both are of radius 1

and length 4. The height of both cylinders lie along the z-axis, and the first cylinder (in white) stays static at the origin, while the other cylinder (in grey), with centre at $z = 4$, assumes different rotations about its principal x-axis. The angle of rotation is from 0 to 360 degrees, with 10 degrees increments. The two cylinders have external contact at the planar faces when $\theta = 0, 180$ and 360 degrees. The DCR of the two cylinders with respect to the $+z$-direction computed by our algorithm is shown in Fig. 10.

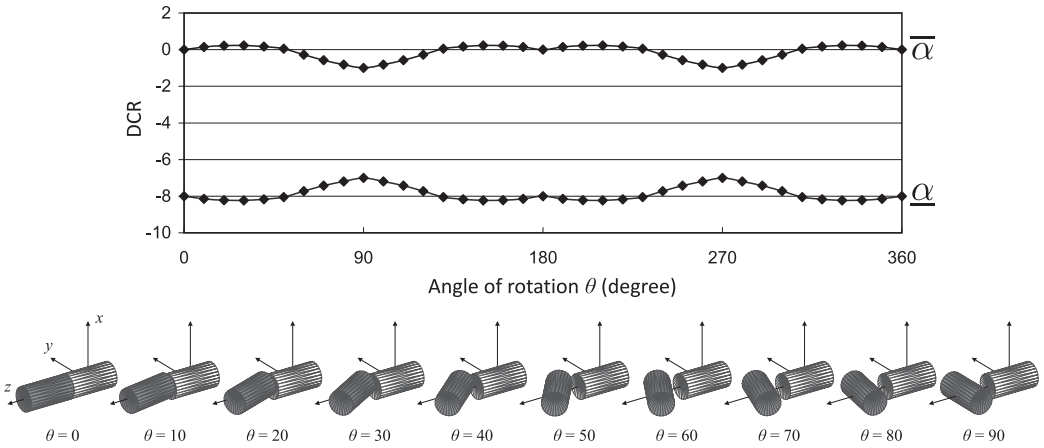


Figure 10: The DCR between two cylinders with respect to the z-direction. The white cylinder is static and the grey cylinder rotates about its principle x-axis. The first 10 instants (with the angle of rotation for the grey cylinder $\theta = 0$ to 90) of the two cylinders are shown.

5. Conclusion

We have presented a novel method for computing the directional contact range (DCR) between two convex polyhedra with respect to a given direction. The DCR of two convex polyhedra can be computed by finding the intersections of a line with the Minkowski difference M of the polyhedra. We consider the problem in the dual space where a face on M corresponds to a vertex on the dual polyhedron M^*, and formulate the DCR computation in forms of searching a vertex which attains the maximum signed distance from a plane. The search problem in the dual space is easily shown to be convex, and a search scheme is devised accordingly to locate the face that contains the required intersection on M efficiently. The search scheme is divided into three stages, each working only on a subset of the faces on M. This division

allows the elimination of most EE-type faces whose worst case complexity is $\mathcal{O}(n^2)$. Our experimental results show that our algorithm exhibits efficient performance. Although our tests do not experience major robustness problems, we note here that the signed distance function $\mathrm{d}(f)$ is non-linear with respect to the distance between the duality centre $\mathbf{c}$ and the intersection of f and $\mathbf{co}$. Possible numerical issues thus induced will be further explored. We shall also investigate the possibility of extending our method to work on a convex decomposition of non-convex M, which can then be used to compute the DCR of two non-convex objects.

Acknowledgement

This work is supported in part by research grants from the Research Grant Council of the Hong Kong SAR (HKU 7031/01E), and in part by the Natural Science Foundation of Shandong Province (Y2005G03).

A. Major concepts

A.1. Duality transformation

There are different formulations for duality transformation. A more general form is to consider the self-dual duality with respect to a given non-singular quadric surface $\mathcal{B} : X^T BX = 0$ where $X = (x, y, z, 1)^T$ is the homogeneous coordinates of a point, and B is a 4×4 real symmetric matrix. The dual of a point Y_0 is a plane $\mathcal{Y} : Y_0^T BX = 0$ (the *polar* of Y_0) and the dual of a plane $\mathcal{V} : V_0^T X = 0$ is a point $U_0 = B^{-1}V_0$ (the *pole* of $\mathcal{V}$ [14]). It is easy to verify that if $\mathcal{Y}$ is the dual of Y_0, then Y_0 is the dual of $\mathcal{Y}$. Also, if Y_0 is a point on $\mathcal{B}$, its dual is the tangent plane to $\mathcal{B}$ at Y_0. In this work, we consider duality transformation with respect to the unit sphere in $\mathbb{E}^3$. The dual relationship between a point and a plane in $\mathbb{E}^3$ in terms of affine coordinates $\mathbf{x} = (x, y, z)^T$ is as follows. Suppose a plane Π, not passing through the origin, is given by $A^T\mathbf{x} = k$ in the *primal space* $\mathbb{E}$, where $A \in \mathbb{R}^3$ and k is a nonzero real number. A *duality transformation* maps Π to a point $\mathbf{w} = A/k$ in the *dual space* $\mathbb{E}^{3*}$. A point $\mathbf{u} \neq \mathbf{0}$ in $\mathbb{E}^3$ is transformed to a plane $\mathcal{U} : \mathbf{u}^T\mathbf{x} = 1$ in $\mathbb{E}^{3*}$. If we extend $\mathbb{E}^3$ to include the plane at infinity (i.e., the extended Euclidean space), a plane passing through the origin in $\mathbb{E}^3$ is mapped to a point at infinity in $\mathbb{E}^{3*}$; whereas the origin in $\mathbb{E}^3$ is transformed to the plane at infinity in $\mathbb{E}^{3*}$. Note that $\mathbb{E}^3$ is the dual space of $\mathbb{E}^{3*}$. We use ψ^* to denote the dual counterpart of an entity ψ in $\mathbb{E}^3$. We may also

consider a duality transformation centred at an arbitrary point $\mathbf{c} \in \mathbb{E}^3$. This can be done by first translating the origin in $\mathbb{E}^3$ to $\mathbf{c}$ before applying duality, and we call $\mathbf{c}$ the *centre of duality*.

A.2. Gaussian image of a polyhedron

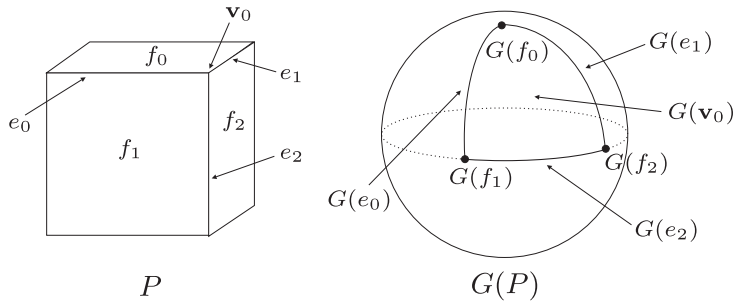


Figure 11: A polyhedron P and its Gaussian image $G(P)$ on $\mathcal{S}^2$.

The Gaussian image $G(P)$ of a convex polyhedron P is a planar graph embedded on the unit sphere $\mathcal{S}^2$ (Fig. 11). The Gaussian image of any feature (i.e., vertex, edge or face) ϕ of a polyhedron P is the set of normal directions of planes that may come into contact with P at ϕ. In other words, ϕ is the supporting feature of P in the directions represented by its Gaussian image. Hence, a face $f \in \mathcal{F}_P$ corresponds to a point $G(f) = \hat{\mathbf{n}}(f) \in \mathcal{S}^2$ where $\hat{\mathbf{n}}(f)$ is the unit normal vector of f; an edge $e \in \mathcal{E}_P$ common to two faces $f_0, f_1 \in P$ corresponds to a great arc $G(e)$ connecting two vertices $G(f_0)$ and $G(f_1)$; a vertex $\mathbf{v} \in \mathcal{V}_P$ common to the faces $f_0, \ldots, f_m$ corresponds to a convex spherical polygon $G(\mathbf{v})$ whose vertices are $G(f_0), \ldots, G(f_m)$.

B. Proofs of correctness of the algorithm

Theorem 1 (Correctness of Search-FV)**.** *The face f_{fv} returned by Search-FV attains the maximum signed distance d_{fv} among all convex faces in $\mathcal{F}_{\text{fv}}$ with respect to $\mathbf{c}$, i.e., $f_{\text{fv}} = \arg\max_f \{\mathrm{d}(f) \mid f \in \mathcal{F}_{\text{fv}} \cap \hat{\mathcal{F}}_{\mathbf{c}}\}$ and $d_{\text{fv}} = \mathrm{d}(f_{\text{fv}})$.*

Proof. Since the initial face f_0 is in $\hat{\mathcal{F}}_{\mathbf{c}}$ and that the signed distance function is increasing over $\hat{\mathcal{F}}_{\mathbf{c}}$, the face f returned by Search-FV must be in $\hat{\mathcal{F}}_{\mathbf{c}}$. Next, we show that f in also in $\mathcal{F}_{\text{fv}}$. Consider the set of faces $\mathcal{F}_{\text{fv}}$ and its corresponding dual $\mathcal{F}^*_{\text{fv}}$. If for every two faces $f_0, f_1 \in \mathcal{F}_P$ that share an edge, we connect $F^*(f_0, \mathrm{s}_{-Q}(f_0))$ and $F^*(f_1, \mathrm{s}_{-Q}(f_1))$ by an edge, then by the construction

of M and the properties of duality, we know that the point set $\mathcal{F}_{\text{fv}}^*$ and the augmented edges form a polyhedron W^*. Since $\mathcal{F}_{\text{fv}}^* \subset \mathcal{V}_{M^*}$ and M^* is convex, W^* must be convex too. Now, Search-FV searches locally for a vertex in W^* that attains the largest signed distance to the plane $\mathbf{o}^*$. The search path also follows the adjacency of the faces in P and therefore is along the edges of W^*. As W^* is convex, the search will eventually stop at a dual vertex f^* of a face $f \in \mathcal{F}_{\text{fv}}$ attaining the maximum signed distance among all dual vertices in $\mathcal{F}_{\text{fv}}^*$ (corresponding to the face set $\mathcal{F}_{\text{fv}}$). □

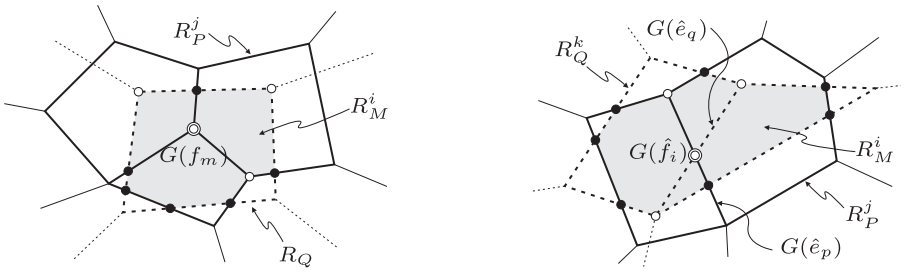
Theorem 2 (Correctness of Search-VF)**.** *The face f_{vf} returned by Search-VF attains the maximum signed distance d_{vf} among all convex faces in $\mathcal{F}_{\text{vf}}$ with respect to $\mathbf{c}$, i.e., $f_{\text{vf}} = \arg\max_f\{\mathrm{d}(f) \mid f \in \mathcal{F}_{\text{vf}} \cap \hat{\mathcal{F}}_{\mathbf{c}}\}$ and $d_{\text{vf}} = \mathrm{d}(f_{\text{vf}})$.*

Proof. Similar to the proof of Theorem 1, by considering the symmetry of P and $-Q$ in the two procedures Search-FV and Search-VF. □

Lemma 3. *The initial face set $\mathcal{FS}_0$ in Search-EE includes all EE-type faces adjacent to f_m, which is the face attaining the maximum signed distance in $(\mathcal{F}_{\text{fv}} \cup \mathcal{F}_{\text{vf}}) \cap \hat{\mathcal{F}}_{\mathbf{c}}$. Moreover, if $f_{\max}$ (i.e., the face with the maximum signed distance in $\hat{\mathcal{F}}_{\mathbf{c}}$) is in $\mathcal{F}_{\text{ee}}$, then the initial face set $\mathcal{FS}_0$ in* Search-EE *must contain at least one convex face (with respect to $\mathbf{c}$) $f_e \in \mathcal{F}_{\text{ee}}$ such that $\mathrm{d}(f_e) > \max\{d_{\text{fv}}, d_{\text{vf}}\}$.*

Proof. Without loss of generality, we assume that $f_m = F\big(f_p, \mathrm{s}_{-Q}(f_p)\big)$ is the starting face in Search-EE with $\mathrm{d}(f_m) = \max\{d_{\text{fv}}, d_{\text{vf}}\}$, where $f_p \in \mathcal{F}_P$, $\mathrm{s}_{-Q}(f_p) \in \mathcal{V}_{-Q}$. The neighbouring faces of f_m on M are those faces that are incident to the vertices of f_m. Consider the Gaussian images $G(M)$, $G(P)$ and $G(-Q)$. Let R_M^i be the neighbouring regions of $G(f_m)$ in $G(M)$, R_P^j be the neighbouring regions of $G(f_p)$ in $G(P)$ and R_Q be the region $G\big(\mathrm{s}_{-Q}(f_p)\big)$ in $G(-Q)$. Hence, the neighbouring faces of f_m correspond to those points defining the regions R_M^i (Fig. 12).

Note that $G(f_m)$ and $G(f_p)$ are the same point on the Gaussian sphere $\mathcal{S}^2$. Since $G(f_p)$ lies inside the region R_Q, R_M^i must be the intersection of R_P^j and R_Q. Therefore, the points of R_M^i must be either (A) the points of R_P^j or R_Q, or (B) the intersections of an arc of R_P^j with an arc of R_Q (Fig. 12(a)). The latter set of points (B) correspond to the face set $\mathcal{FS}_0$ in Search-EE. If $\mathcal{FS}_0$ is empty, the neighbouring faces of f_m can only be faces corresponding to points in set (A), i.e., the faces in $\mathcal{F}_{\text{fv}} \cup \mathcal{F}_{\text{vf}}$. Then f_m has the maximum signed distance among all its neighbours, since $\mathrm{d}(f_m) = \max\{d_{\text{fv}}, d_{\text{vf}}\}$. On



(a) (b)

Figure 12: The Gaussian map of M with $G(P)$ and $G(-Q)$ shown in solid and dotted lines, respectively, (a) for showing the neighbouring faces of $f_m = F\big(f_p, \mathrm{s}_{-Q}(f_p)\big)$. Thick solid lines marked R_P^j, the neighbouring regions of $G(f_p)$ in $G(P)$, thick dotted lines marked $R_Q = G\big(\mathrm{s}_{-Q}(f_p)\big)$ in $G(-Q)$, and the grey regions are R_M^i, the neighbouring regions of $G(f_m)$ in $G(M)$. The black (FV- or VF-types) and white (EE-type) points are the neighbouring faces of f_m; (b) for showing the neighbouring faces of $\hat{f}_i = F(\hat{e}_p, \hat{e}_q)$. Thick solid line marked R_P^j, the neighbouring regions of $G(\hat{e}_p)$ in $G(P)$, thick dotted lines marked R_Q^k, the neighbouring regions of $G(\hat{e}_q)$ in $G(-Q)$, and the grey regions are R_M^i, the neighbouring regions of $G(\hat{f}_i)$ in $G(M)$. The black (FV- or VF-types) and white (EE-type) points are the neighbouring faces of $\hat{f}_i$.

the other hand, if $\mathcal{FS}_0$ is non-empty, and all faces $f_e \in \mathcal{FS}_0$ are such that $\mathrm{d}(f_e) < \mathrm{d}(f_m)$. Again, f_m has the maximum signed distance among all its neighbours. In both cases, it implies that $f_m = f_{\max}$. However, this contradicts that $f_{\max}$ is in $\mathcal{F}_{\mathrm{ee}}$. Hence, there must be at least a face $f_e \in \mathcal{FS}_0$ such that $\mathrm{d}(f_e) > \max\{d_{\mathrm{fv}}, d_{\mathrm{vf}}\}$. Moreover, since f_m is convex with respect to $\mathbf{c}$ and $\mathrm{d}(f_e) > \mathrm{d}(f_m)$, due to the convexity of $\mathrm{d}(\cdot)$, f_e must also be a convex face. □

Lemma 4. *The face set $\mathcal{FS}_{i+1}$ in Search-EE contains all EE-type faces that are adjacent to the face $\hat{f}_i = F(\hat{e}_p, \hat{e}_q)$, where $\hat{e}_p \in \mathcal{E}_P$ and $\hat{e}_q \in \mathcal{E}_{-Q}$.*

Proof. The neighbouring faces of $\hat{f}_i$ on M are those faces incident to the vertices of $\hat{f}_i$. Consider the Gaussian images $G(M)$, $G(P)$ and $G(-Q)$. The point $G(\hat{f}_i)$ is the intersection of the two arcs $G(\hat{e}_p)$ and $G(\hat{e}_q)$ (Fig. 12(b)). Let R_M^i be the neighbouring regions of $G(\hat{f}_i)$, R_P^j be the two neighbouring regions of $G(\hat{e}_p)$ and R_Q^k be the two neighbouring regions of $G(\hat{e}_q)$. The regions R_M^i must be the intersection of R_P^j and R_Q^k; hence, the points defining R_M^i must be the intersections of the arcs of R_P^j and R_Q^k, and also some points from R_P^j, R_Q^k. Hence the faces in $\mathcal{FS}_{i+1}$ in Search-EE corresponds to the

intersections of the arcs of R_P^j and R_Q^k, which are all the EE-type neighbours of $\hat{f}_i$. □

Theorem 5 (Correctness of SEARCH-EE)**.** *The face f_{ee} returned by Search-EE attains the maximum signed distance among all convex faces in $\hat{\mathcal{F}}_{\mathbf{c}}$, i.e., $f_{\text{ee}} = f_{\max} = \arg\max_f\{\mathrm{d}(f) \mid f \in \hat{\mathcal{F}}_{\mathbf{c}}\}$.*

Proof. The face $f_{\max} = \arg\max_f\{\mathrm{d}(f) \mid f \in \hat{\mathcal{F}}_{\mathbf{c}}\}$ with the maximum signed distance $d_{\max}$ must be in either $\mathcal{F}_{\text{fv}} \cup \mathcal{F}_{\text{vf}}$ or $\mathcal{F}_{\text{ee}}$. Suppose $f_{\max}$ is in $\mathcal{F}_{\text{fv}} \cup \mathcal{F}_{\text{vf}}$, then $f_{\max}$ equals either f_{fv} or f_{vf} returned by Search-FV or Search-VF, respectively. Also, no face in $\mathcal{F}_{\text{ee}}$ has a larger signed distance than $d_{\max}$. In this case, the flow of Search-EE guarantees that $f_{\max}$ is returned.

Now, suppose that $f_{\max}$ is in $\mathcal{F}_{\text{ee}}$. Lemma 3 proves that the initial face set $\mathcal{FS}_0$ is not empty and that $\max\{\mathrm{d}(f) \mid f \in \mathcal{FS}_0\} > \max\{d_{\text{fv}}, d_{\text{vf}}\}$. Hence, the iteration in Search-EE will proceed. For each iteration $i > 0$, d_{ee} is the signed distance of the current face, and $\mathcal{FS}_i$ is the set of all EE-type faces neighbouring to the current face (by Lemma 4). The iteration stops when the signed distance of the current face is the maximum among all its neighbouring EE-type faces. Since d_{ee} is increasing for each iteration, $d_{\text{ee}} > \max\{d_{\text{fv}}, d_{\text{vf}}\}$ also means that the signed distance of the current face is the maximum among all its neighbouring FV- and VF-type faces. Hence, Search-EE stops at a face f_{ee} in $\mathcal{F}_{\text{ee}}$ with a maximum signed distance among all its neighbouring faces in M. Since the initial face f_m is convex with respect to $\mathbf{c}$, and d_{ee} is increasing for each iteration, due to the convexity of $\mathrm{d}(\cdot)$, f_{ee} attains the maximum signed distance among all convex faces in $\hat{\mathcal{F}}_{\mathbf{c}}$. □

References

[1] P. Jiménez, F. Thomas, C. Torras, 3D collision detection: a survey., Computers & Graphics 25 (2) (2001) 269–285.

[2] M. C. Lin, D. Manocha, Collision and proximity queries, in: Handbook. of Discrete and Computational Geometry, 2003.

[3] S. Cameron, R. Culley, Determining the minimum translational distance between two convex polyhedra., in: Proc. IEEE Int. Conf. on Robotics and Automation, 1986, pp. 591–596.

[4] M. C. Lin, J. Canny, A fast algorithm for incremental distance calculation., in: Proc. IEEE Int. Conf. on Robotics and Automation, Sacremento, April, 1991, pp. 1008–1014.

[5] B. Mirtich, V-Clip: Fast and robust polyhedral collision detection., ACM Trans. Graph. 17 (3) (1998) 177–208.

[6] E. G. Gilbert, D. W. Johnson, S. S. Keerthi, A fast procedure for computing the distance between objects in three-dimensional space, IEEE J. Robot. Automat. RA-4 (1988) 193–203.

[7] P. K. Agarwal, L. J. Guibas, S. Har-Peled, A. Rabinovitch, M. Sharir, Penetration depth of two convex polytopes in 3D., Nordic Journal of Computing 7 (3) (2000) 227–240.

[8] Y. J. Kim, M. C. Lin, D. Manocha, Incremental penetration depth estimation between convex polytopes using dual-space expansion., IEEE Trans. Visual. Comput. Graphics 10 (2) (2004) 152–163.

[9] D. P. Dobkin, J. Hershberger, D. G. Kirkpatrick, S. Suri, Computing the intersection-depth of polyhedra., Algorithmica 9 (6) (1993) 518–533.

[10] O. Günther, Efficient structures for geometric data management, Springer-Verlag New York, Inc., New York, NY, USA, 1988.

[11] I. Kolingerová, Convex polyhedron-line intersection detection using dual representation, The Visual Computer 13 (1) (1997) 42–49.

[12] B. Grünbaum, Convex Polytopes, Wiley, 1967.

[13] D. P. Dobkin, D. G. Kirkpatrick, A linear algorithm for determining the separation of convex polyhedra., J. Algorithms 6 (3) (1985) 381–392.

[14] H. Levy, Projective and Related Geometry, Macmillan, 1964.