Collision Detection for Unions of Convex Bodies With Smooth Boundaries Using Closed-Form Contact Space Parameterization

Sipu Ruan, *Member, IEEE*, Xiaoli Wang, *Student Member, IEEE*, and Gregory S. Chirikjian, *Fellow, IEEE*

***Abstract*—This paper studies the narrow phase collision detection problem for two general unions of convex bodies encapsulated by smooth surfaces. The approach, namely *CFC* (Closed-Form Contact space), is based on parameterizing their contact space in closed-form. The first body is dilated to form the contact space while the second is shrunk to a point. Then, the collision detection is formulated as finding the closest point on the parametric contact space with the center of the second body. Numerical solutions are proposed based on the point-to-surface distance as well as the common-normal concept. Furthermore, when the two bodies are moving or under linear deformations, their first time of contact is solved continuously along the time-parameterized trajectories. Benchmark studies are conducted for the proposed algorithms in terms of solution stability and computational cost. Applications of the sampling-based motion planning for robot manipulators are demonstrated.**

***Index Terms*—Collision avoidance, computational geometry, motion and path planning.**

I. Introduction

COLLISION detection is one of the core modules in many fields such as robot motion planning [1], multi-body dynamics [2] and simulations [3]. Its main objectives are: (a) determining the contact status of two bodies; (b) computing minimum distance if separated, or predicting maximum penetration depth if in contact; and (c) recording the corresponding witness information (*e.g.,* closest points, contact normal vectors, etc). For two moving bodies, especially at fast speeds, a continuous collision detection (CCD) algorithm is also able to return their first time of contact.

Hundreds of algorithms have been proposed to make collision detection and proximity queries more efficient [4], [5], most of which consider polyhedral bodies. The performance of a collision detection for this type of bodies relies heavily on the surface complexity.

As an alternative to polyhedra, convex bodies with smooth bounding surfaces that can be expressed in implicit or parametric forms are also popular in modelling objects. Typical examples include ellipsoids, superquadrics, poly-ellipsoid, etc [6]. The collision detection algorithms between two such bodies are mostly formulated as optimization problems. For example, proximity distance between bodies bounded by implicit surfaces can be computed by convex optimization when they are static or under translational motions [7]. But the penetration depth can not be obtained when in contact. Another set of objective functions is based on a necessary condition of minimum distance, *i.e.,* the *common-normal* concept [8]–[10]. The normal vectors at the nearest points are anti-parallel with each other as well as with the line connecting these two nearest points. Using this setting as the objective function has achieved great accuracy and efficiency during the optimization process. However, its extension to the continuous case is unclear.

Recently, the exact *contact space* between two convex bodies with smooth boundaries has been computed in closed form [11]. The solution provides parametric expressions of the center of one body when touching the other externally. In the static case, the collision detection can be formulated as computing the minimum distance from the center of one body to the contact space. The extension to the continuous case is also clear by adding an extra time parameter. By using the proposed closed-form expression under linear deformations, the contact space can be updated efficiently along the time-parameterized trajectory of motions. Utilizing the advantages of the closed-form contact space expression, this paper proposes novel solutions to the narrow-phase collision queries between two unions of convex bodies with smooth boundaries. The major contributions of this paper are:

- A unified framework for static collision detection and penetration depth computations as well as continuous collision detection is proposed using the closed-form contact space parameterization;
- The problems are formulated as point-to-surface distance queries via nonlinear optimization;
- Continuous collision detection is solved by using the dynamic closed-form expression and adding a time variable to the optimization objectives.

Manuscript received 23 February 2022; accepted 20 June 2022. Date of publication 13 July 2022; date of current version 26 July 2022. This letter was recommended for publication by Associate Editor J. Yu and Editor H. Kurniawati upon evaluation of the reviewers' comments. This work was supported in part by National Research Foundation, Singapore, under its Medium Sized Centre Programme - Centre for Advanced Robotics Technology Innovation (CARTIN), under Subaward R-261-521-002-592, in part by Singapore MOE Tier 1 under Grant R-265-000-655-114, in part by the National University of Singapore (NUS) Startup under Grants R-265-000-665-133 and R-265-000-665-731, and in part by NUS Faculty Board under Grant C-265-000-071-001. *(Corresponding author: Gregory S. Chirikjian.)*

The authors are with the Department of Mechanical Engineering, National University of Singapore, Singapore 119077 (e-mail: ruansp@nus.edu.sg; wangxiaoli@u.nus.edu; mpegre@nus.edu.sg).

This letter has supplementary downloadable material available at https://doi.org/10.1109/LRA.2022.3190629, provided by the authors.

Digital Object Identifier 10.1109/LRA.2022.3190629

II. Literature Review

This section reviews related work on collision detection for bodies enclosed by polyhedra and smooth surfaces.

A. General Collision Detection Methods

One of the classic algorithms to compute collision and distance queries between two convex polyhedra is *Gilbert-Johnson-Keerthi (GJK)* [12], [13]. The algorithm is based on the Minkowski operations of two convex polytopes. It iteratively generates "simplex" using the support functions of the convex polyhedra. To further compute penetration depth between two convex polytopes, the *expanding polytope algorithm (EPA)* [14] is proposed. A more comprehensive review on Minkowski operations applying to collision detection can be referred to [15]. Another type of methods is based on *bounding volume hierarchy (BVH)* [16], which uses geometric primitives such as spheres, axis-aligned bounding boxes (AABB) or oriented bounding boxes (OBB) [17] to encapsulate polytopes. To further deal with multiple objects in the space, a hierarchical data structure like *octree* [18] is proposed. Many of these well-known algorithms are implemented in the Flexible Collision Library (FCL) [19].

B. Collision Detection Algorithms for Convex Bodies With Smooth Boundaries

One method that calculates the proximity distance for general implicit surfaces is proposed in [7]. The proximity queries are formulated as a convex constrained optimization problem, which is solved by the interior-point method. CCD is also addressed for translational movements, which preserves the convexity of the problem. For ellipsoids, collision status can be inferred from their characteristic polynomial algebraically [20]. The *algebraic separation condition (ASC)* gives the exact status as to whether two ellipsoids are separated or in collision, which also applies to the continuous case [21]. These algorithms show great success in collision detection between two ellipsoids. However, they cannot compute distance or extend to other geometric types.

There are various methods proposed for querying collision status and distance between superquadrics. One method is based on the *common-normal* concept. In [8], the common-normal condition is formulated as a 2-dimensional unconstrained optimization problem. In [22], a constrained optimization is formulated based on the implicit functions and alignment of the normal vectors. Furthermore, in [23], the anti-parallelism of the normal vectors is numerically solved using Newton-Raphson method with analytical Jacobian matrices. Benchmark studies on distance computation for superquadrics under nonlinear tapering deformation (*i.e.,* superovoids) are conducted [9]. Recently, the normal parameterization of a surface has been applied into collision detection [10]. An efficient and accurate fixed-point iteration method is proposed to solve for the normal alignment objective, which inspires one method proposed in this paper.

Moreover, poly-ellipsoids generalize the single ellipsoidal model by combining different one-eighth ellipsoids in the eight octants, formulating a more general non-symmetric model [24]. Further extension to a poly-superquadric model is proposed recently [25], which applies the *midway-point* method to solve for the contact queries.

Another type of surface, using sum-of-squares (SOS) polynomials, has also been shown to be efficient in modeling the environment and performing collision detection by optimizations [26]. By changing the degrees of the polynomials, various smooth surfaces can be obtained to tightly enclose a body. More recent work has shown effectiveness of this model in motion planning [27], [28].

III. Mathematical Preliminary

A. Contact Space for Two General Bodies

Consider two bodies S_1 and S_2 embedded in $\mathbb{R}^N$, the contact space is an $(N-1)$-dimensional manifold embedded in $\mathbb{R}^N$, which traces a reference point when the two bodies externally touch each other. It is closely related to the concept of Minkowski sums of the two bodies, defined as

$$S_1 \oplus S_2 \doteq \{\mathbf{x}_1 + \mathbf{x}_2 \,|\, \mathbf{x}_1 \in S_1, \mathbf{x}_2 \in S_2\}. \tag{1}$$

In practice, the boundary of Minkowski sums is of more interest, denoted as $\partial(S_1 \oplus S_2)$. Using this definition, the contact space can be defined mathematically as

$$\Sigma \doteq \partial[S_1 \oplus (-S_2)], \tag{2}$$

where $(-S_2)$ is the reflection of S_2 about the origin of its body frame.

B. Implicit and Parametric Surfaces

In the general N-dimensional Euclidean space, a surface can be written in either implicit or parametric forms, or both of them. Given a body $S \subset \mathbb{R}^N$, the implicit form of its boundary surface is denoted as $\Psi(\mathbf{x}) = 0$, where $\mathbf{x} \in \mathbb{R}^N$ is a point in the Euclidean space. This is an inside-outside scalar function. The point on the surface can also be expressed in parametric form, *i.e.,* $\mathbf{x} \doteq \mathbf{f}(\boldsymbol{\psi}) \in \mathbb{R}^N$, where $\boldsymbol{\psi} \in \mathbb{R}^{N-1}$ is the parameter set of the surface. Concretely for a 3D convex superquadric model, the implicit expression is

$$\Psi_{\mathrm{SQ}}(\mathbf{x}) = \left(\left(\frac{x_1}{a}\right)^{2/\epsilon_2} + \left(\frac{x_2}{b}\right)^{2/\epsilon_2}\right)^{\epsilon_2/\epsilon_1} + \left(\frac{x_3}{c}\right)^{2/\epsilon_1} - 1 \tag{3}$$

where $\mathbf{x} \doteq [x_1, x_2, x_3]^T$, a, b, c are its semi-axes lengths and $\epsilon_1, \epsilon_2 \in (0, 2)$ are the exponents that ensure strict convexity.

C. Gradient Parameterization of Smooth Surfaces Enclosing Convex Bodies

Using the implicit expressions, the outward-pointing gradient can be directly computed as

$$\mathbf{m} \doteq \nabla_{\mathbf{x}} \Psi(\mathbf{x}). \tag{4}$$

Normalizing the gradient gives the outward normal vector, *i.e.,* $\mathbf{n} \doteq \frac{\nabla_{\mathbf{x}}\Psi(\mathbf{x})}{\|\nabla_{\mathbf{x}}\Psi(\mathbf{x})\|_2}$, where $\|\cdot\|_2$ denotes the vector 2-norm. The condition that the bounding surface ∂S is smooth guarantees the existence of the gradient and the denominator is non-zero. This is also called the Gauss map of a surface. If the body S

is also convex (*i.e.,* all principal curvatures are positive at all points on the surface), its inverse Gauss map exists and is unique. This means that each normal vector is able to parameterize a unique point on the surface of the body. This is also referred here as the *normal-parameterization* of a surface, *i.e.,* $\mathbf{x} = \widetilde{\mathbf{f}}(\mathbf{n})$. In this case, (4) can also be inverted, resulting in the *gradient-parameterization* of a surface, *i.e.,* $\mathbf{x} = \widetilde{\widetilde{\mathbf{f}}}(\mathbf{m})$.

In the case of a superquadric,

$$\widetilde{\widetilde{\mathbf{f}}}_{\mathrm{SQ}}(\mathbf{m}) = \begin{pmatrix} a\left(\frac{a\epsilon_1}{2}m_1\right)^{\epsilon_2/(2-\epsilon_2)}\left[\gamma(m_3)\right]^{(\epsilon_1-\epsilon_2)/(2-\epsilon_2)} \\ b\left(\frac{b\epsilon_1}{2}m_2\right)^{\epsilon_2/(2-\epsilon_2)}\left[\gamma(m_3)\right]^{(\epsilon_1-\epsilon_2)/(2-\epsilon_2)} \\ c\left(\frac{c\epsilon_1}{2}m_3\right)^{\epsilon_1/(2-\epsilon_1)} \end{pmatrix}, \tag{5}$$

where $m_j = \mathbf{m}\cdot\mathbf{e}_j$ and $\gamma(m_3) = 1 - (\frac{c\epsilon_1}{2}m_3)^{2/(2-\epsilon_1)}$ [11]. A nice property of a superquadric is the dual relationships between its surface point and surface gradient. When the superquadric is defined by parameters $\{a, b, c, \epsilon_1, \epsilon_2\}$, its gradient also satisfies the superquadric model with parameters $\{2/(a\,\epsilon_1), 2/(b\,\epsilon_1), 2/(c\,\epsilon_1), 2-\epsilon_1, 2-\epsilon_2\}$, which is denoted throughout this paper as SQ′.

D. Closed-Form Contact Space Using Gradient Parameterization

The contact space between S_1 and S_2 that is parameterized by the gradient $\mathbf{m}_1$ of S_1 can be computed in closed form as

$$\mathbf{x}_\Sigma(\mathbf{m}_1) = \widetilde{\widetilde{\mathbf{f}}}_1(\mathbf{m}_1) - \widetilde{\widetilde{\mathbf{f}}}_2\left(-\frac{\Phi(\mathbf{m}_1)}{\|\mathbf{m}_1\|_2}\mathbf{m}_1\right), \tag{6}$$

where $\Phi(\mathbf{m}_1) = \left\|\mathbf{g}_2\left(\frac{\mathbf{g}_2^{-1}(\mathbf{m}_1)}{\|\mathbf{g}_2^{-1}(\mathbf{m}_1)\|_2}\right)\right\|_2$ [11]. For the superquadric,

$$\mathbf{g}_2(\mathbf{u}) = \begin{pmatrix} \frac{2}{a_2\epsilon_{21}}\,u_1^{2-\epsilon_{22}}\left(u_1^2+u_2^2\right)^{\frac{\epsilon_{22}-\epsilon_{21}}{2}} \\ \frac{2}{b_2\epsilon_{21}}\,u_2^{2-\epsilon_{22}}\left(u_1^2+u_2^2\right)^{\frac{\epsilon_{22}-\epsilon_{21}}{2}} \\ \frac{2}{c_2\epsilon_{21}}\,u_3^{2-\epsilon_{21}} \end{pmatrix}, \tag{7}$$

where $a_2, b_2, c_2, \epsilon_{21}$ and ϵ_{22} are the shape parameters of S_2, and $u_j = \mathbf{u}\cdot\mathbf{e}_j$ with $\|\mathbf{u}\|_2 = 1$. The inverse function of (7) can be obtained as

$$\mathbf{g}_2^{-1}(\mathbf{m}_1) = \begin{pmatrix} \left(\frac{a_2\epsilon_{21}}{2}m_{11}\right)^{\frac{1}{2-\epsilon_{22}}}\left[\rho(m_{21},m_{22})\right]^{\frac{\epsilon_{21}-\epsilon_{22}}{4-2\epsilon_{21}}} \\ \left(\frac{b_2\epsilon_{21}}{2}m_{12}\right)^{\frac{1}{2-\epsilon_{22}}}\left[\rho(m_{21},m_{22})\right]^{\frac{\epsilon_{21}-\epsilon_{22}}{4-2\epsilon_{21}}} \\ \left(\frac{c_2\epsilon_{21}}{2}m_{13}\right)^{\frac{1}{2-\epsilon_{21}}} \end{pmatrix}, \tag{8}$$

where $m_{ij} = \mathbf{m}_i\cdot\mathbf{e}_j$ is the j-th entry of the gradient of S_i and $\rho(m_{21},m_{22}) = (\frac{a_2\epsilon_{21}}{2}m_{21})^{2/(2-\epsilon_{22})} + (\frac{b_2\epsilon_{21}}{2}m_{22})^{2/(2-\epsilon_{22})}$.

If the two bodies are deformed by arbitrary linear transformation matrices M_1 and M_2 respectively, their contact space can be computed in closed-form as

$$\mathbf{x}'_\Sigma(\mathbf{m}_1) = M_1\,\widetilde{\widetilde{\mathbf{f}}}_1(\mathbf{m}_1) - M_2\,\widetilde{\widetilde{\mathbf{f}}}_2\left(-\frac{\Phi(M_2^T\,M_1^{-T}\mathbf{m}_1)}{\|M_2^T\,M_1^{-T}\mathbf{m}_1\|_2}M_2^T\,M_1^{-T}\mathbf{m}_1\right). \tag{9}$$

The linear transformations include rotation, shearing, scaling, etc. An example of the closed-form contact space between a superquadric and poly-ellipsoid under random linear transformations is demonstrated in Fig. 1.

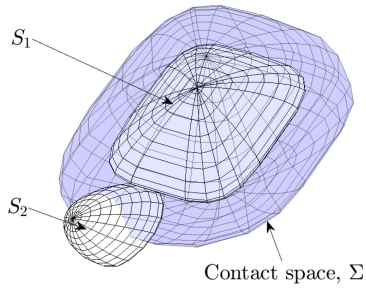


Fig. 1. Closed-form contact space between S_1 (superquadric) and S_2 (poly-ellipsoid) in general poses. The contact space Σ from (9) is shown in transparent blue color.

IV. Problem Formulation

In general, the minimum proximity distance between two bodies S_1 and S_2 can be computed by solving

$$\min_{\mathbf{x}_1,\mathbf{x}_2} \|\mathbf{x}_1 - \mathbf{x}_2\|_2, \text{ subject to } \mathbf{x}_1 \in S_1,\, \mathbf{x}_2 \in S_2\,. \tag{10}$$

The two bodies are in contact when $S_1 \cap S_2 \neq \emptyset$, and (10) will result in zero. Using Minkowski sums, *i.e.,* $S_1 \oplus (-S_2)$, an equivalent formulation can be obtained as

$$\min_{\mathbf{x}} \|\mathbf{p}_2 - \mathbf{x}\|_2, \text{ subject to } \mathbf{x} \in S_1 \oplus (-S_2)\,, \tag{11}$$

where $\mathbf{p}_2$ is the center of S_2. The two bodies are in contact when $\mathbf{p}_2 \in S_1 \oplus (-S_2)$. Consequently, the objective function becomes zero. Both (10) and (11) are able to return minimum distance when two bodies are separated and zero distance when two bodies are in contact. However, they cannot provide penetration depth information.

A. The Contact Space Formulation

Using the contact space concept in (2), the minimum distance problem can be alternatively formulated as

$$\min_{\mathbf{x}_\Sigma} \|\mathbf{p}_2 - \mathbf{x}_\Sigma\|_2, \text{ subject to } \mathbf{x}_\Sigma \in \Sigma\,. \tag{12}$$

When the two bodies are separated or just touch each other, this formulation is equivalent to both (10) and (11). In addition, when the two bodies interpenetrate, their penetration depth can also be computed using this formulation.

A necessary condition of the optimization problem in (12) is to use the *common normal* concept [8]: the outward-pointing normal vector at $\mathbf{x}_\Sigma$ is anti-parallel to the vector connecting $\mathbf{x}_\Sigma$ and $\mathbf{p}_2$. Using this concept, another optimization can be formulated to solve for the minimum distance or penetration depth problem as

$$\min_{\mathbf{x}_\Sigma} \|\mathbf{m}(\mathbf{x}_\Sigma) \times (\mathbf{p}_2 - \mathbf{x}_\Sigma)\|_2, \text{ subject to } \mathbf{x}_\Sigma \in \Sigma\,, \tag{13}$$

where $\times$ denotes the cross product of two vectors. Note that this is just a necessary condition of (12) since the common normal condition can also be satisfied when the point $\mathbf{p}_2$ and the contact space Σ have maximal distance.

B. The Continuous Case

For the continuous collision detection, the goal is to search for the first time of contact if the two bodies will collide along the whole motion or the minimal separation distance when no collision will occur. The optimization setting in (12) can be extended to solve continuous collision detection (CCD) problems easily using the contact space formulation. Similar to the static case, a general optimization problem for CCD can be formulated as

$$\begin{aligned} \min_{\mathbf{x}_\Sigma,\tau} \ & \|\mathbf{p}_2(\tau) - \mathbf{x}_\Sigma(\tau)\|_2 \\ \text{subject to } & \mathbf{x}_\Sigma(\tau) \in \Sigma(\tau) \doteq \partial[S_1(\tau) \oplus (-S_2(\tau))] \\ & \tau \in [0, \tau_{\max}] \,, \end{aligned} \tag{14}$$

where τ is the additional time parameter to be optimized, $S_i(\tau)$ $(i = 1, 2)$ and $\Sigma(\tau)$ are the transformed body and the corresponding contact space at τ. The range of τ is set a-priori. If the velocities of the two objects are given, then a maximum travel time is also required. On the other hand, if the start and goal poses of the two objects are given, we can set the range to be [0,1]. In both cases, poses of the two objects at each intermediate time step can be computed using interpolation techniques.

V. Collision Detection Based on Parameterized Closed-Form Contact Space

This section proposes methods to solve for the collision detection problem formulated in Section IV in both static and continuous cases. The outward-pointing gradient vector $\mathbf{m}$ is parameterized by angular parameters, *i.e.,* $\mathbf{m} = \mathbf{m}(\boldsymbol{\psi})$. Using the same parameters, surface points on the contact space can also be explicitly defined, *i.e.,* $\mathbf{x}_\Sigma(\mathbf{m}(\boldsymbol{\psi}))$.

A. Collision Detection in the Static Case

Three methods are proposed to solve for the collision detection queries in the static case.

1) Minimizing Point-to-Surface Distance: The first algorithm solves (12) using a nonlinear least squares optimization as

$$\min_{\boldsymbol{\psi}} \ \frac{1}{2} \|\mathbf{p}_2 - \mathbf{x}_\Sigma(\mathbf{m}(\boldsymbol{\psi}))\|_2^2 \,. \tag{15}$$

A trust-region dogleg algorithm is used to solve for this problem, which applies the Ceres solver [29].

2) Nonlinear Optimization Using Common-Normal Concept: The second method for the static case uses the common normal concept (as in (13)), which is formulated as

$$\min_{\boldsymbol{\psi}} \ \frac{1}{2} \|\mathbf{m}(\boldsymbol{\psi}) \times [\mathbf{p}_2 - \mathbf{x}_\Sigma(\mathbf{m}(\boldsymbol{\psi}))]\|_2^2 \,. \tag{16}$$

The same trust-region algorithm using Ceres solver is used to solve this problem.

3) Derivative-Free Fixed-Point Iteration Method: The third method also applies the common normal concept but uses a fixed-point iteration method, which does not require a derivative. At each iteration of the algorithm, the gradient is updated using the fixed point function based on the previous two iterations as

$$\begin{aligned} \widetilde{\widetilde{\mathbf{r}}}_{i+1} &= \mathbf{p}_2 - \mathbf{x}_\Sigma(\mathbf{m}_i) + \\ &\quad \frac{\left(\mathbf{n}_i - (\mathbf{n}_i^T \mathbf{n}_{i-1})\mathbf{n}_{i-1}\right)^T \left(\mathbf{x}_\Sigma(\mathbf{m}_i) - \mathbf{x}_\Sigma(\mathbf{m}_{i-1})\right)}{\|\mathbf{n}_i \times \mathbf{n}_{i-1}\|_2^2} \mathbf{n}_i \,, \\ \mathbf{m}_{i+1} &= \mathbf{h}\left(\widetilde{\widetilde{\mathbf{r}}}_{\mathbf{i+1}}\right) \,, \end{aligned} \tag{17}$$

where $\mathbf{n}_j = \mathbf{m}_j / \|\mathbf{m}_j\|_2$ $(j = i-1, i)$. The initial values require two gradient vectors, which are chosen as

$$\begin{cases} \mathbf{m}_0 = \mathbf{h}(\mathbf{p}_2) \,, \\ \mathbf{m}_1 = \mathbf{h}\left(\frac{\mathbf{m}_0}{\|\mathbf{m}_0\|_2} + \frac{\mathbf{p}_2 - \mathbf{x}_\Sigma(\mathbf{m}_0)}{\|\mathbf{p}_2 - \mathbf{x}_\Sigma(\mathbf{m}_0)\|_2}\right) \,, \end{cases} \tag{18}$$

where $\mathbf{h} : \mathbb{R}^N \to \mathbb{R}^N$ maps any directional vector $\mathbf{v}$ into the gradient $\mathbf{m}$. Specifically for the superquadric, it can be computed analytically using the dual relationships between the surface point and gradient (as discussed in Section III-C). Observe that $\Psi_{\mathrm{SQ}'}(k\mathbf{m}) + 1 = k^{2/(2-\epsilon_1)}(\Psi_{\mathrm{SQ}'}(\mathbf{m}) + 1)$, where k is a constant scalar and $\Psi_{\mathrm{SQ}'}(\mathbf{m}) = 0$. Then, any vector $\mathbf{v} = k\mathbf{m}$ along the line of $\mathbf{m}$ satisfies this equation. Directly calculating k gives

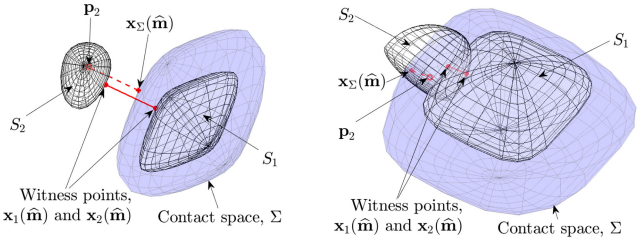
$$\mathbf{m} \doteq \mathbf{h}(\mathbf{v}) = \frac{\mathbf{v}}{\left(1 + \Psi_{\mathrm{SQ}'}(\mathbf{v})\right)^{\frac{2-\epsilon_1}{2}}} \,. \tag{19}$$

Note that, it is possible to compute $\mathbf{h}(\cdot)$ for other geometric types, which remains further investigation.

The idea for (17) is interpreted from the geometric point of view as follows [10]. Given $\mathbf{m}_{i-1}$ and $\mathbf{m}_i$, the point on the line of $\mathbf{m}_i$ that is closest to $\mathbf{m}_{i-1}$ can be computed as $\widetilde{\mathbf{p}}_{i+1}$. Its expression is the third term of $\widetilde{\mathbf{r}}_{i+1}$ in (17). Then, $\widetilde{\mathbf{r}}_{i+1}$ is computed as the vector pointing from $\widetilde{\mathbf{p}}_{i+1}$ to $\mathbf{p}_2$, which is finally mapped to $\mathbf{m}_{i+1}$ via $\mathbf{h}(\cdot)$.

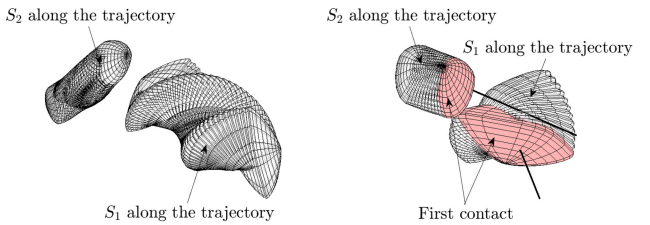
The optimization terminates when the change of the gradient vector is smaller than a tolerance δ_{tol}, *i.e.,* $\|\mathbf{m}_{i+1} - \mathbf{m}_i\| < \delta_{\mathrm{tol}}$. In this work, $\delta_{\mathrm{tol}} \doteq 10^{-12}$ is used.

Fig. 2 shows the results from the collision queries in two scenarios: separated (Fig. 2(a)) and colliding (Fig. 2(b)). For the case of separation, the witness points, *i.e.,* $\mathbf{x}_i(\widehat{\mathbf{m}})$ $(i = 1, 2)$, are defined as the closest points on the two bodies. And for the case of in-collision, the witness points are the ones on the different bodies that have the deepest penetrated distance. For both cases, the witness points are computed from the minimum



(a) Minimum distance when not in contact. (b) Penetration depth when colliding.

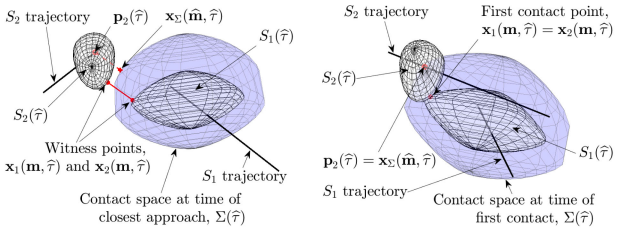
Fig. 2. Demonstrations of the proposed collision detection algorithm in static case.



(a) The whole motions of S_1 and S_2 when not colliding.

(b) The partial motions of S_1 and S_2, stopped at the first contact.

Fig. 3. Demonstrations of two scenarios that the proposed continuous collision detection algorithm can handle.



(a) Minimum distance when not in contact during the whole motions.

(b) Time of first contact when colliding in the middle of motions.

Fig. 4. Solutions of the cases in Fig. 3 using the proposed algorithm for CCD based on the closed-form contact space.

distance from $\mathbf{p}_2$ to the contact space Σ and parameterized by the optimal parameter $\widehat{\mathbf{m}}$.

B. Continuous Collision Detection Using Nonlinear Least-Squares Method

The idea of optimizing the proximity distance between a point and contact space surface can be extended to solve continuous collision detection (CCD) problems. Similar to the static case, a general optimization problem for CCD can be formulated as

$$\min_{\boldsymbol{\psi},\tau} \frac{1}{2} \left\| \mathbf{p}_2(\tau) - \mathbf{x}_\Sigma(\mathbf{m}(\boldsymbol{\psi}),\tau) \right\|_2^2 , \tag{20}$$

where $\tau \in [0, \tau_{\max}]$ is the time parameter to be optimized simultaneously. Fig. 3 demonstrates two scenarios of the proposed algorithm can handle: separated along the whole motions and contact in the middle. And Fig. 4 shows the corresponding solutions using the proposed algorithm for the scenarios presented in Fig. 3.

When the two bodies are separated along the whole motions (Fig. 3(a)), the distance of their closest approach is computed (Fig. 4(a)). And $\mathbf{x}_i(\widehat{\mathbf{m}},\widehat{\tau})$ $(i=1,2)$ is the corresponding witness point on each body. The center of S_2 at the optimal solution, *i.e.,* $\mathbf{p}_2(\widehat{\tau})$, is outside the contact space $\Sigma(\widehat{\tau})$. On the other hand, when the two bodies collide while moving (Fig. 3(b)), the information of their first contact is returned (Fig. 4(b)). In this case, $\mathbf{p}_2(\widehat{\tau}) = \mathbf{x}_\Sigma(\widehat{\mathbf{m}},\widehat{\tau}) \in \Sigma(\widehat{\tau})$ and the witness point on each body $\mathbf{x}_1(\widehat{\mathbf{m}},\widehat{\tau}) = \mathbf{x}_2(\widehat{\mathbf{m}},\widehat{\tau})$ is the first contact point. This CCD setting might result in multiple solutions of the time variable when (20) is zero. In our formulation, there will be at most two discrete common points of the trajectory and contact space since this is a line-surface intersection problem. Then, the solution with smaller value of the time parameter is the one at the first time-of-impact.

To solve the problem numerically, we formulate it as a nonlinear least-squares problem and solve it using the trust-region method that is similar to those in the static case. For a general linear transformation that includes rotation, shearing and scaling, the contact space expression, *i.e.,* $\mathbf{x}_\Sigma(\mathbf{m}(\boldsymbol{\psi}),\tau)$, is updated using (9) at each iteration of the optimization process.

C. Initial Values for the Nonlinear Optimization

Unlike (11), which is formulated as a convex optimization, the proposed optimization problems including (12), (13) and (14) are not convex. Therefore, the nonlinear optimization solvers only guarantee local optimality, and they are sensitive to initial conditions. Instead of randomly choosing initial variables, this work uses the angular parameter of S_2 center as viewed in the body frame of S_1, denoted as ${}^1\mathbf{p}_2 \doteq [{}^1p_{2,1}, {}^1p_{2,2}, {}^1p_{2,3}]^T$. In the 3D static case,

$$\boldsymbol{\psi}_0 = \left\{ \text{atan2}\left({}^1p_{2,3}, \sqrt{{}^1p_{2,1}^2 + {}^1p_{2,2}^2} \right), \text{atan2}\left({}^1p_{2,2}, {}^1p_{2,1} \right) \right\} .$$

As for the fix-point iteration method, the initial value is selected using (18). In the continuous case, the additional time parameter τ is initialized as $\tau_0 = 0$.

VI. Benchmark Study

This section evaluates the performance of the proposed collision detection framework using the closed-form contact space. Different geometric primitives are used, *i.e.,* ellipsoids (E), superquadrics (SQ) and polyellipsoids (PE). The three proposed methods in the static case and the method in continuous case are compared, along with some existing state-of-the-art collision detection algorithms designed for convex bodies enclosed by smooth surface. For a reference, the well-known method for discrete convex polyhedra is also included in the benchmark studies, where the surface of each body is discretized as mesh. All the benchmark studies are implemented in C++ and performed in an Intel Xeon CPU at 2.80GHz.

A. Benchmark Settings

For each pair of bodies, a total of 10^4 trials of collision detection are conducted. In the static case, each experimental trial is set by one pair of two bodies with random shapes and poses. In the continuous case, the shapes of the two bodies, and their initial and goal poses are randomly generated. The motion primitives include: (a) pure translation and (b) a combination of translation and rotation.

The algorithms in the comparisons are summarized in Table I. Our proposed algorithms are labeled as *CFC-[Cost]-[Solver]*, where *[Cost]* denotes the objective type in the cost function, *i.e.,* using distance (*Dist*) or common-normal (*CN*), and *[Solver]* denotes the solver type, *i.e.,* least-squares (*LS*) or fixed-point iteration method (*FP*).

In order to make the comparisons as fair as possible, we re-implemented and modified the algorithms that are designed for

TABLE I
ALGORITHMS INVOLVED IN THE BENCHMARK STUDY. OUR PROPOSED ALGORITHMS ARE SHOWN IN BOLD

Algorithm notation	Method	Solver
FCL	GJK [12]	FCL [19]
Implicit	[7]	Interior-point (IP), IFOpt [30]
CN-LS	Eq. (21)	Trust-region (TR), Ceres [29]
CN-FP	[10]	Fixed-point (FP) iteration
CFC-Dist-LS	Eq. (15) / (20)	Trust-region (TR), Ceres [29]
CFC-CN-LS	Eq. (16)	Trust-region (TR), Ceres [29]
CFC-CN-FP	Eq. (17)	Fixed-point (FP) iteration

convex bodies bounded by smooth surfaces, i.e., *Implicit* [7], *CN-LS* [9] and *CN-FP* [10], to incorporate with the formulations in this paper. In particular, for *Implicit*, the optimization settings are the same with [7], but it is solved by IFOpt [30] – an open-source wrapper for the IPOpt [31] implementation of interior-point optimization algorithm. For *CN-LS*, the cost function is slightly modified from [9], [23] into

$$\min_{\psi_1,\psi_2} \frac{1}{2} \left\| \begin{pmatrix} \mathbf{n}_1(\boldsymbol{\psi}_1) \times \mathbf{n}_2(\boldsymbol{\psi}_2) \\ \mathbf{n}_1(\boldsymbol{\psi}_1) \times [\mathbf{f}_1(\boldsymbol{\psi}_1) - \mathbf{f}_2(\boldsymbol{\psi}_2)] \end{pmatrix} \right\|_2^2, \tag{21}$$

where $\boldsymbol{\psi}_i \, (i = 1, 2)$ is the angular parameter of each body surface. The modification is aligned with the formulation in (16), but preserves the idea of matching the direction of the normal and the vector connecting surface points. The nonlinear least-squares optimization is also solved by the same Ceres solver using trust-region dogleg algorithm. For *CN-FP*, the fixed-point iteration that updates normal vectors is the same with that in [10]. But the normal parameterization is modified from the generating potential into a direct computation of the inverse Gauss map $\hat{\mathbf{f}}(\mathbf{n})$. For *FCL*, the surfaces of the two bodies are discretized into meshes using 100 sampled points.

B. Evaluation Metrics

The evaluation metrics for the benchmarks include accuracy based on the necessary condition and efficiency based on computational time and number of iterations. A collision detection result is acceptable only if

$$\|\widehat{\mathbf{n}} \times (\widehat{\mathbf{x}}_1 - \widehat{\mathbf{x}}_2)\|_2 < \delta \,, \tag{22}$$

where $\widehat{\mathbf{x}}_1, \widehat{\mathbf{x}}_2, \widehat{\mathbf{n}}$ are the optimal solutions of the witness point on S_1, S_2 and the corresponding normal vector respectively and $\delta \geq 0$ is set to be 10^{-5}. Note that this accuracy metric is also valid for the continuous case since when the two bodies contact each other, $\widehat{\mathbf{x}}_1 = \widehat{\mathbf{x}}_2$ which results in zero value of the metric; and when the two bodies are separated, this metric also necessarily gives the minimum distance condition.

C. Results and Analysis

The comparison results are presented in both static and continuous cases. Different geometric types are used mutually in the comparisons.

1) The Static Case: The benchmark of computational time for different geometric pairs are shown in Fig. 5. The unit of computational time is micro seconds (μs). For the sake of clarity in demonstration, the averaged time spent on the collision detection between one typical SQ-PE pair is shown here in Table II. The accuracy metric based on (22) is evaluated for each optimization-based algorithms, which is shown in Table III. The average numbers of iterations (rounded to the closest integer) are compared in Table IV. The methods using the same type of solver are compared together for fairness.

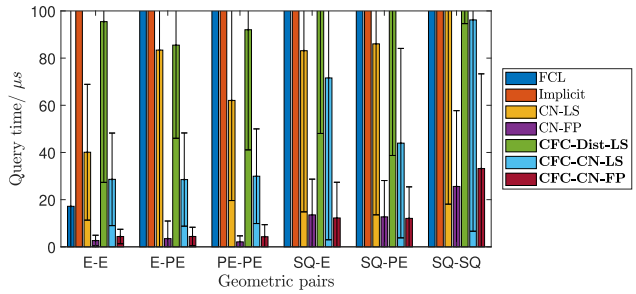


Fig. 5. Collision query time for two static bodies using different algorithms. The computational time above $100\mu s$ is trimmed for clearer demonstration. The black error bars indicate the standard deviation of the querying time.

TABLE II
AVERAGED COMPUTATIONAL TIME FOR STATIC COLLISION DETECTION BETWEEN SQ AND PE

Algorithm	Initial (μs)	Query (μs)	Total (μs)
FCL	6905	9844	16748
Implicit	237.5	37843	38080
CN-LS	0.7092	86.0	86.8
CN-FP	**0.2309**	12.7	12.9
CFC-Dist-LS	0.8912	117.0	117.9
CFC-CN-LS	0.6029	44.0	44.6
CFC-CN-FP	0.3033	**12.1**	**12.4**

The best performance among all the algorithms is shown in bold.

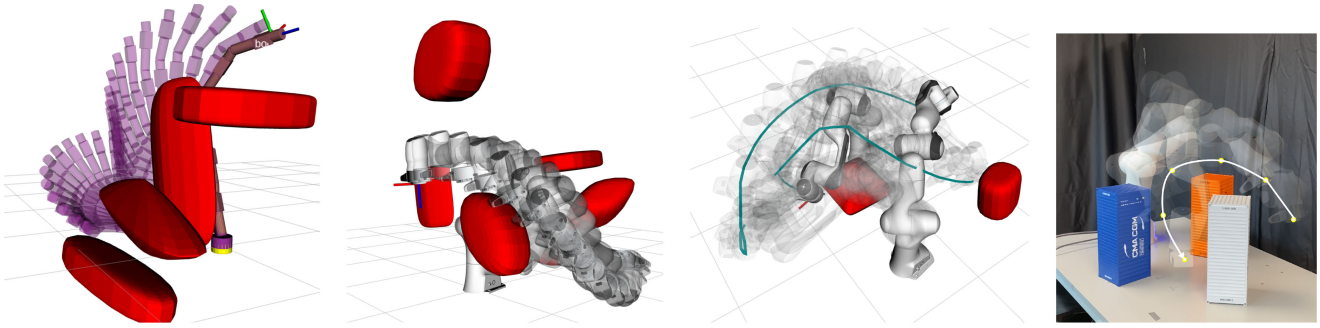
TABLE III
ACCURACY (%) OF STATIC COLLISION DETECTION FOR DIFFERENT PAIRS OF GEOMETRIC MODELS

Algorithm	E-E	E-PE	PE-PE	SQ-E	SQ-PE	SQ-SQ
Implicit	97.64	97.42	96.70	72.23	70.95	65.65
CN-LS	97.43	97.57	97.63	91.99	91.86	85.26
CN-FP	**99.94**	99.89	99.83	96.05	96.23	92.43
CFC-Dist-LS	99.85	99.71	**99.85**	**97.93**	**97.09**	**92.74**
CFC-CN-LS	99.54	98.91	98.90	95.67	94.94	91.65
CFC-CN-FP	**99.94**	**99.94**	99.84	96.00	96.17	92.46

The best performance among all the algorithms is shown in bold.

TABLE IV
NUMBER OF ITERATIONS IN SOLVING THE NONLINEAR OPTIMIZATION PROBLEMS FOR STATIC COLLISION DETECTION. OUR PROPOSED ALGORITHMS ARE SHOWN IN BOLD

Algorithm	E-E	E-PE	PE-PE	SQ-E	SQ-PE	SQ-SQ
CN-LS	6	6	6	9	9	14
CFC-Dist-LS	23	23	22	24	24	25
CFC-CN-LS	5	6	5	7	7	9
CN-FP	10	10	11	16	16	21
CFC-CN-FP	9	9	10	15	15	20



(a) 15-DOF snake-like robot. (b) Single Panda robot arm. (c) Panda robot dual-arm system. (d) Physical experiment.

Fig. 6. Demonstration of the sampling-based motion planning using the proposed collision detection algorithm. (a) 15-DOF snake-like robot. (b) Single Panda robot arm. (c) Panda robot dual-arm system. (d) Physical experiment.

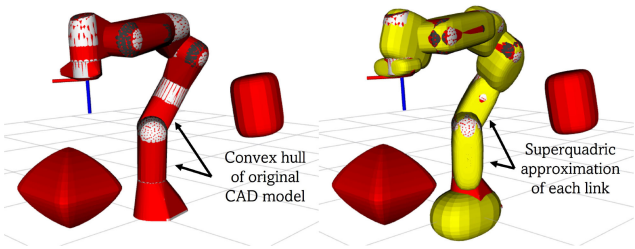


Fig. 7. The convex hull of the original CAD model and fitted superquadric model for each link of the Panda robot.

TABLE V
COMPARISON RESULTS FOR THE CONTINUOUS CASE BETWEEN SQ AND PE UNDER DIFFERENT MOTION TYPES

Motion type	Initial time (μs)	Query time (μs)	Total time (μs)	Number iteration	Accuracy
Trans.	1.0	231.6	232.6	30	90.80%
Rot. + Trans.	1.1	296.8	297.9	33	84.90%

Among the proposed algorithms based on the closed-form contact space, optimizing the cost functions using the common-normal concept is more efficient than that using the point-to-surface distance. It is clear that, under the same least-squares solving framework, the *CFC-CN-LS* algorithm runs much faster and converges more quickly. But in term of the accuracy, the *CFC-Dist-LS* algorithm outperforms *CFC-CN-LS* in most trials. For both the methods that optimize the common-normal cost, *CFC-CN-FP* further speeds up the computation due to the simpler updates of the gradient vectors in each iteration. It is also able to achieve a higher accuracy than *CFC-CN-LS*.

The comparisons with other state-of-the-art optimization-based collision detection algorithms show the advantage of using the proposed contact space formulation. For the same least-squares setting, the proposed *CFC-CN-LS* algorithm runs around half of the time than the *CN-LS* algorithm. The accuracy is also higher than the latter. For *CFC-CN-FP* and *CN-FP* algorithms, which both use the fixed-point method in each iteration, the computational time and accuracy are competitive to each other. On the one hand, *CN-FP* requires computing the normal updates using the equation like (14) twice, one for each body. But *CFC-CN-FP* only compute (14) once per iteration. On the other hand, our computation of $\mathbf{x}_\Sigma$ is more complex than theirs, which only needs to compute the normal parameterization of the surface point. By combining these two factors, the two methods perform similar in the benchmark studies.

2) The Continuous Case: In the continuous case, the performance of our proposed algorithm *CFC-Dist-LS* for different motion types is compared. Two cases, including pure translation and rigid motion, are studied. Table V shows the benchmark results of CCD between the SQ-PE pair.

The computational time for the case of rigid motions is larger than that of a pure translation, *i.e.,* an averaged increase of around 25%. But the averaged number of iterations only has slight difference, meaning that the time spent on each iteration increases but the algorithm still converges well. The increased time at each iteration is mainly due to the more complex computations of interpolation. Comparing to the static case, the query time in the CCD case doubles, with an increased number of iterations. This means that the optimization problem becomes more complex by the additional time variable, but still converges in an acceptable rate and accuracy.

VII. APPLICATION

In this section, the proposed *CFC-CN-FP* collision detector is implemented as a sub-module in the sampling-based motion planning algorithms from the *Open Motion Planning Library (OMPL)* [32]. The RRT-connect planner [33] is used to plan the motion. In the case studies, a 7-DOF Franka Emika Panda robot manipulator and a 15-DOF snake-like robot are used. The rigid links of the manipulators are encapsulated by superquadrics and the environment is filled with unions of superquadric obstacles.

1) Motion Planning for a Snake-Like Robot: In this case study (Fig. 6(a)), a 15-DOF snake-like robot arm is designed virtually. Each link is defined by a superquadric model a priori.

2) Motion Planning for a Single Franka Emika Panda Robot Manipulator: In this case (Fig. 6(b)), the original mesh representation of each link of the Panda robot is encapsulated by the superquadric model, using the recently proposed fitting

method [34]. The convex hull of the CAD model of each link and the resulting superquadric model are demonstrated in Fig 7.

3) Motion Planning for a Dual-Arm System With Two Panda Robots: A dual arm system is constructed by two Panda manipulators (Fig. 6(c)) in this case study. The problem is solved in a centralized way, where the individual configuration space of the two arms are concatenated together, formulating a 14-dimensional configuration space.

4) Physical Experiments Using the Panda Robot: Physical experiments are also conducted using a single Franka Panda robot (Fig. 6(d)). The planning environment is constructed by some random objects on the table, which has some narrow regions. The scene is captured by a depth camera and proceeded into point cloud data. The point cloud is then clustered into disjoint sets, each of which is fitted by a superquadric model using [34]. The planning process is the same as in Section VII-A2. The resulting joint trajectory is fed into the control module of the manipulator to execute motions.

VIII. Conclusion

This paper proposes a collision detection framework for unions of convex bodies with smooth boundary. It is based on parameterizing the contact space of the two bodies in closed-form using the outward-pointing gradient. The distance between one body center and the contact space is solved using nonlinear optimization techniques in both static and continuous cases. In the future, the authors will investigate the possibility of formulating contact space under nonlinear deformation in closed-form.

References

[1] H. Heidari and M. Saska, "Collision-free trajectory planning of multi-rotor UAVs in a wind condition based on modified potential field," *Mech. Mach. Theory*, vol. 156, 2021, Art. no. 104140.

[2] E. Corral, R. G. Moreno, M. G. García, and C. Castejón, "Nonlinear phenomena of contact in multibody systems dynamics: A review," *Nonlinear Dyn.*, vol. 104, pp. 1269–1295, 2021.

[3] C. Li, M. Tang, R. Tong, M. Cai, J. Zhao, and D. Manocha, "P-cloth: Interactive complex cloth simulation on multi-GPU systems using dynamic matrix assembly and pipelined implicit integrators," *ACM Trans. Graph.*, vol. 39, no. 6, pp. 1–15, 2020.

[4] P. Jiménez, F. Thomas, and C. Torras, "3D collision detection: A survey," *Comput. Graph.*, vol. 25, no. 2, pp. 269–285, 2001.

[5] E. Åblad, D. Spensieri, R. Bohlin, and A.-B. Strömberg, "Continuous collision detection of pairs of robot motions under velocity uncertainty," *IEEE Trans. Robot.*, vol. 37, no. 5, pp. 1780–1791, Oct. 2021.

[6] A. H. Barr, "Superquadrics and angle-preserving transformations," *IEEE Comput. Graph. Appl.*, vol. 1, no. 1, pp. 11–23, Jan. 1981.

[7] N. Chakraborty, J. Peng, S. Akella, and J. E. Mitchell, "Proximity queries between convex objects: An interior point approach for implicit surfaces," *IEEE Trans. Robot.*, vol. 24, no. 1, pp. 211–220, Feb. 2008.

[8] C. Wellmann, C. Lillie, and P. Wriggers, "A contact detection algorithm for superellipsoids based on the common-normal concept," *Eng. Comput.*, vol. 25, no. 5, pp. 432–442, 2008.

[9] A. A. Gonçalves, A. Bernardino, J. Jorge, and D. S. Lopes, "A benchmark study on accuracy-controlled distance calculation between superellipsoid and superovoid contact geometries," *Mechanism Mach. Theory*, vol. 115, pp. 77–96, 2017.

[10] U. J. Römer, A. Fidlin, and W. Seemann, "The normal parameterization and its application to collision detection," *Mechanism Mach. Theory*, vol. 151, 2020, Art. no. 103906.

[11] S. Ruan and G. S. Chirikjian, "Closed-form minkowski sums of convex bodies with smooth positively curved boundaries," *Comput.-Aided Des.*, vol. 143, 2021, Art. no. 103133.

[12] E. G. Gilbert, D. W. Johnson, and S. S. Keerthi, "A fast procedure for computing the distance between complex objects in three-dimensional space," *IEEE J. Robot. Autom.*, vol. 4, no. 2, pp. 193–203, Apr. 1988.

[13] M. Montanari, N. Petrinic, and E. Barbieri, "Improving the GJK algorithm for faster and more reliable distance queries between convex objects," *ACM Trans. Graph.*, vol. 36, no. 3, pp. 1–17, 2017.

[14] G. Van Den Bergen, "Proximity queries and penetration depth computation on 3D game objects," *Game Developers Conf.*, vol. 170, 2001.

[15] W. Cox, L. While, and M. Reynolds, "A review of methods to compute minkowski operations for geometric overlap detection," *IEEE Trans. Vis. Comput. Graph.*, vol. 27, no. 8, pp. 3377–3396, Aug. 2021.

[16] J.-W. Chang, W. Wang, and M.-S. Kim, "Efficient collision detection using a dual Obb-sphere bounding volume hierarchy," *Comput.-Aided Des.*, vol. 42, no. 1, pp. 50–57, 2010.

[17] S. Gottschalk, M. C. Lin, and D. Manocha, "OBBTree: A hierarchical structure for rapid interference detection," in *Proc. 23rd Annu. Conf. Comput. Graph. Interactive Techn.*, 1996, pp. 171–180.

[18] D. Jung and K. K. Gupta, "Octree-based hierarchical distance maps for collision detection," *J. Robotic Syst.*, vol. 14, no. 11, pp. 789–806, 1997.

[19] J. Pan, S. Chitta, and D. Manocha, "FCL: A general purpose library for collision and proximity queries," in *Proc. IEEE Int. Conf. Robot. Automat.*, 2012, pp. 3859–3866.

[20] W. Wang, J. Wang, and M.-S. Kim, "An algebraic condition for the separation of two ellipsoids," *Comput. Aided Geometric Des.*, vol. 18, no. 6, pp. 531–539, 2001.

[21] Y.-K. Choi, J.-W. Chang, W. Wang, M.-S. Kim, and G. Elber, "Continuous collision detection for ellipsoids," *IEEE Trans. Vis. Comput. Graph.*, vol. 15, no. 2, pp. 311–325, Mar./Apr. 2009.

[22] R. Fontes Portal, J. Pereira Dias, and L. Gonçalves de Sousa, "Contact detection between convex superquadric surfaces," *Arch. Mech. Eng.*, vol. 57, no. 2, pp. 165–186, 2010.

[23] D. S. Lopes, M. T. Silva, J. A. Ambrósio, and P. Flores, "A mathematical framework for rigid contact detection between quadric and superquadric surfaces," *Multibody Syst. Dyn.*, vol. 24, no. 3, pp. 255–280, 2010.

[24] J. F. Peters, M. A. Hopkins, R. Kala, and R. E. Wahl, "A poly-ellipsoid particle for non-spherical discrete element method," *Eng. Comput.*, vol. 26, no. 6, pp. 645–657, 2009.

[25] S. Wang and S. Ji, "Poly-superquadric model for DEM simulations of asymmetrically shaped particles," *Comput. Part. Mechanics*, vol. 9, pp. 299–313, 2021.

[26] A. A. Ahmadi, G. Hall, A. Makadia, and V. Sindhwani, "Geometry of 3D environments and sum of squares polynomials," in *Proc. Robotics: Sci. Syst.*, Cambridge, Massachusetts, Jul. 2017, doi: 10.15607/RSS.2017.XIII.071.

[27] B. El Khadir, J. B. Lasserre, and V. Sindhwani, "Piecewise-linear motion planning amidst static, moving, or morphing obstacles," in *Proc. IEEE Int. Conf. Robot. Automat.*, 2021, pp. 7802–7808.

[28] J. Guthrie, M. Kobilarov, and E. Mallada, "Closed-form minkowski sum approximations for efficient optimization-based collision avoidance," 2022, *arXiv:2203.15977*.

[29] S. Agarwal, K. Mierle, and Others, "Ceres solver," Accessed: Jun. 07, 2021. http://ceres-solver.org

[30] A. W. Winkler, "Ifopt - a. modern, light-weight, eigen-based C++ interface to Nonlinear Programming solvers Ipopt and Snopt," 2018. [Online]. Available: https://doi.org/10.5281/zenodo.1135046

[31] A. Wächter and L. T. Biegler, "On the implementation of an interior-point filter line-search algorithm for large-scale nonlinear programming," *Math. Program.*, vol. 106, no. 1, pp. 25–57, 2006.

[32] I. A. Sucan, M. Moll, and L. E. Kavraki, "The open motion planning library," *IEEE Robot. Automat. Mag.*, vol. 19, no. 4, pp. 72–82, Dec. 2012.

[33] J. J. Kuffner and S. M. LaValle, "RRT-connect: An efficient approach to single-query path planning," in *Proc. IEEE Int. Conf. Robot. Automat.*, 2000, pp. 995–1001.

[34] W. Liu, Y. Wu, S. Ruan, and G. S. Chirikjian, "Robust and accurate superquadric recovery: A probabilistic approach," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2021, pp. 2676–2685.