

Collision Detection between Smooth Convex Bodies via Riemannian Optimization Framework

Seoki An, Somang Lee, Jeongmin Lee, Sunkyung Park and Dongjun Lee

Abstract—Collision detection is a fundamental problem across various fields such as robotics, physical simulation, and computer graphics. While numerous studies have provided efficient solutions, based on the well-known Gilbert, Johnson, and Keerthi (GJK) algorithm and Expanding Polytope Algorithm (EPA), existing methods utilizing GJK-EPA often struggle with smooth strictly convex shapes like ellipsoids. This paper proposes a novel approach to the collision detection problem converting it to a problem compatible with an unconstrained Riemannian optimization problem. Moreover, we presents a specific method of solving the problem based on twice differentiable support functions and the Riemannian trust region (RTR) method. The method exhibits fast and robust convergence rate, leveraging the well-established theory of Riemannian optimization. The evaluation studies comparing our method to GJK-EPA method are done with pre-defined primitive shapes. Additionally, a test result with several more complex shapes is demonstrated exhibiting the method’s effectiveness and applicability.

I. INTRODUCTION

Collision detection has earned great interests from robotics, computer graphics, and video games, and many algorithms were developed to efficiently solve it. The most widely adopted collision detection algorithm is the Gilbert, Johnson, and Keerthi (GJK) algorithm [1] which determines whether a collision occurs between two convex bodies. As an extension of GJK algorithm, the expanding polytope algorithm (EPA) [2] further computes the penetration depth and the direction of separation. Both GJK algorithm and EPA utilize the notion of support function [3], which is deeply related to convex sets mathematically. GJK-EPA method has been proved to be efficient and robust with mesh-based geometric representations, and many additional studies have been conducted to enhance the algorithm [4], [5], [6]. Nonetheless, the iteration number of GJK-EPA method significantly increases when dealing with scenarios associated with smooth strictly convex shapes such as ellipsoids due to the discrete nature of its algorithm. Such limitations can pose challenges to simulation or manipulation planning with various real world objects which often exhibit such smooth geometric characteristics.

Recently, other collision detection methods are actively researched based on optimization formulation. Montaut et. al. [7] demonstrated that GJK algorithm can be regarded

This research was supported by the National Research Foundation (NRF) grant funded by the Ministry of Science and ICT (MSIT) of Korea (RS-2022-00144468), the Ministry of Trade, Industry & Energy (MOTIE) of Korea (RS-2024-00419641), and HD Hyundai XiteSolution. The authors are with the Department of Mechanical Engineering, IAMD and IOER, Seoul National University, Seoul 08826, Republic of Korea. {seoki97s, hopelee, ljmlgh, sunk1136, djlee}@snu.ac.kr

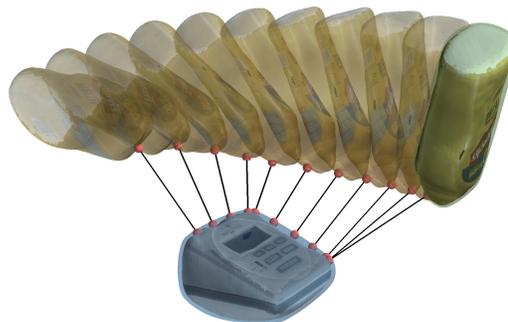


Fig. 1. A snapshot for continuous collision detection between two convex models fitted to the object models from YCB dataset [10] with smooth surfaces. In the figure, the fitted models are scaled larger than the original ones for a clear visualization.

as a specific case of the Frank-Wolfe algorithm in convex optimization. Leveraging this insight, they proposed an accelerated version of GJK algorithm, yet, still retaining fundamental limitations of the original GJK algorithm. Ruan et. al. [8] presented several nonlinear least square formulations and associated algorithms for the collision detection with parameterized superquadric representations. Although the optimization problems introduced in [8] can be solved successfully in many cases, not much theoretical studies for the convergence have been conducted yet, which is important since one failure case can cause significantly different results in simulation or planning applications involving contacts. Finally, in the work of Lee et. al. [9], a novel smooth representation of convex bodies is introduced, with differentiability and strict convexity as while also capable to approximate non-smooth polytopes. Combined with this novel representation, the differentiable contact features can be obtained and the collision detection problem is solved by the Newton-Rhapson method exploiting the derivatives of the obtained contact features in [9]. However, the method in [9] requires additional iterative initialization process for stable convergence in practice.

In this paper, we introduce a novel collision detection method which is capable of computing the contact features (e.g., a contact normal and witness points) between smooth strictly convex bodies while theoretically-guaranteeing fast convergence and robustness in both penetrating and non-penetrating scenarios. We discovered that the collision detection problem is compatible with an unconstrained Riemannian optimization problem. More precisely, among various methods in optimization on manifolds, we adopt the Riemannian trust region (RTR) method [11] to leverage

the smoothness of the convex bodies and ensure a fast convergence rate when applied to the collision detection problem. The major contributions of this paper are twofold:

- To the best of our knowledge, this paper is the very first result to formulate the collision detection as the Riemannian optimization problem; and
- We theoretically establish the global convergence and the local superlinear convergence rate with RTR method by leveraging the second order information from given geometries.

We believe that the algorithm presented in this paper is promising for a wide range of applications in robotics such as simulation [12], motion planning [13], trajectory optimization [14], and even real time control associated with contact and collision avoidance.

The rest of the paper is organized as follows. Mathematical preliminaries are briefly introduced in Sec. II. Then, we describe how the collision detection problem can be reformulated as a Riemannian optimization problem and solved efficiently with RTR method assuring the local superlinear convergence up to the small penetration depth between the two convex bodies in the collision detection problem in Sec. III. Further, differentiable support functions suited to our method are presented in Sec. IV. Exploiting the presented support functions, performance evaluation of our method compared to GJK-EPA method is presented in Sec. V. Lastly, conclusion and future works are discussed in Sec. VI.

II. PRELIMINARY

A. Support Function

Given a convex set \mathcal{C} in \mathbb{R}^n , a support function $h_{\mathcal{C}} : \mathbb{R}^n \rightarrow \mathbb{R}$ for \mathcal{C} is defined by

$$h_{\mathcal{C}}(x) = \sup_{s \in \mathcal{C}} x \cdot s \quad (1)$$

where $x \in \mathbb{R}^n$ and \cdot denotes the standard inner product on \mathbb{R}^n . The limit point $s(x) \in \partial\mathcal{C}$ of $s \in \mathcal{C}$ satisfying $h_{\mathcal{C}}(x) = x \cdot s(x)$ can be interpreted as the farthest point in the direction of x among the points in the closure of \mathcal{C} , and it is often called a support point. In addition, we establish some basic properties of the support function. For any convex sets $A, B \subset \mathbb{R}^n$ and the corresponding support functions h_A, h_B , the following holds:

$$h_A(\lambda x) = \lambda h_A(x), \quad \forall \lambda \geq 0, x \in \mathbb{R}^n \quad (2)$$

$$h_{A+B}(x) = h_A(x) + h_B(x) \quad (3)$$

where $A + B = \{a + b \in \mathbb{R}^n \mid a \in A, b \in B\}$ denotes a Minkowski sum of two sets A, B .

Assuming only closed convex sets, the correspondences between convex sets and support functions defined by (1) are one-to-one [3], which encourages us to exploit support functions as representations for convex bodies. To ease our discussion, any convex bodies without additional statements are considered to be compact (i.e. closed and bounded).

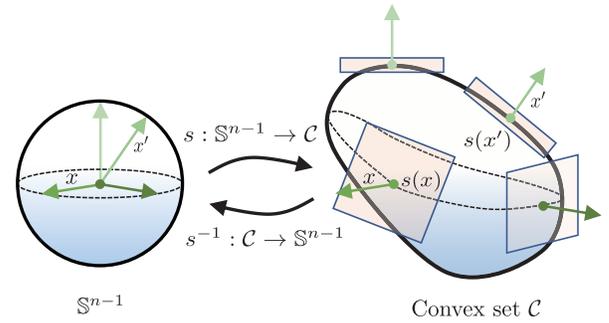


Fig. 2. The map $x \mapsto s(x)$, whose domain is a unit sphere, is an inverse of the Gauss map [16], where the Gauss map assigns each point $p \in \mathcal{C}$ to the unit normal vector x of \mathcal{C} at p .

On the other hand, if $h_{\mathcal{C}}$ is differentiable on $\mathbb{R}^n \setminus \{0\}$, it is known that the support point $s(x)$ can be computed by

$$\frac{\partial h_{\mathcal{C}}(x)}{\partial x} = s(x) \in \mathbb{R}^n.$$

if and only if the convex body \mathcal{C} is strictly convex and x represents a normal vector of \mathcal{C} at $s(x)$ [15]. From the positive homogeneity property (2), we can see that $s(x)$ is uniquely determined by the direction represented by x . This fact implies that restricting the domain of the map $x \mapsto s(x)$ to a sphere makes the map $x \mapsto s(x)$ bijective. Particularly, restricting the domain to a unit $(n-1)$ -sphere \mathbb{S}^{n-1} transforms the map $x \mapsto s(x)$ into an inverse of a Gauss map which has a special role when dealing with the curvature of smooth surfaces in differential geometry [15], [16], [17]. Lastly, note that if $h_{\mathcal{C}}$ is a C^2 function (i.e. twice continuously differentiable) we call \mathcal{C} be of class C^2 .

B. Minimum Translational Distance Model

Contact features between two convex bodies are mainly composed of a penetration depth, a contact normal, and witness points. The contact features may have different mathematical definitions depending on which contact model adopted. One of the most widely adopted method for determining the contact features between two convex bodies is the minimum translational distance (MTD) model [18], [1], [2]. The MTD model defines a function $\text{MTD}^+(\mathcal{C}_1, \mathcal{C}_2)$ for convex bodies $\mathcal{C}_1, \mathcal{C}_2 \subset \mathbb{R}^n$ as follows:

$$\text{MTD}^+(\mathcal{C}_1, \mathcal{C}_2) = \min_{t \in \mathbb{R}^n} \left\{ \|t\| \mid \begin{array}{l} \partial\mathcal{C}_1 \cap \partial(\mathcal{C}_2 + t) \neq \emptyset, \\ \text{Int } \mathcal{C}_1 \cap \text{Int}(\mathcal{C}_2 + t) = \emptyset \end{array} \right\}.$$

where $\text{Int } \mathcal{C}_i$ and $\partial\mathcal{C}$ denote the interior and the boundary of \mathcal{C}_i respectively for each $i = 1, 2$. Then, to distinguish the cases whether a penetration occurs or not, the MTD itself is defined by

$$\text{MTD}(\mathcal{C}_1, \mathcal{C}_2) = \begin{cases} \text{MTD}^+(\mathcal{C}_1, \mathcal{C}_2) & \text{if } \mathcal{C}_1 \cap \mathcal{C}_2 = \emptyset \\ -\text{MTD}^+(\mathcal{C}_1, \mathcal{C}_2) & \text{otherwise} \end{cases}.$$

Remark 1: Computing the MTD function coincides to finding the closest point on $\partial(\mathcal{C}_2 + (-\mathcal{C}_1))$ to the origin, where $\mathcal{C}_2 + (-\mathcal{C}_1)$ denotes the Minkowski sum and $-\mathcal{C}_1 = \{-x \in \mathbb{R}^n \mid x \in \mathcal{C}_1\}$.

Algorithm 1 Basic Form of the Riemannian Optimization

 Initialize $x_0 \in \mathcal{M}$.

repeat

 Compute step $v_k \in T_{x_k} \mathcal{M}$

 Update $x_{k+1} = \text{rt}_{x_k}(v_k)$
until termination condition is satisfied

C. Riemannian Optimization

Suppose that we are given an unconstrained optimization problem

$$\min_{x \in \mathcal{M}} f(x) \quad (4)$$

where \mathcal{M} is a Riemannian manifold with a Riemannian metric $\langle \cdot, \cdot \rangle$ which defines an inner product $\langle \cdot, \cdot \rangle_x : T_x \mathcal{M} \times T_x \mathcal{M} \rightarrow \mathbb{R}$ on a tangent space $T_x \mathcal{M}$ at each $x \in \mathcal{M}$. A class of such problem is called an optimization on manifolds or Riemannian optimization [19]. To deal with its non-linearity, Riemannian optimization exploits a retraction map which is a differentiable map $\text{rt} : \bigcup_{x \in \mathcal{M}} T_x \mathcal{M} \rightarrow \mathcal{M}$, $(x, v) \mapsto \text{rt}_x(v)$ such that each curve $c(t) = \text{rt}_x(tv)$ satisfies $c(0) = x$ and $c'(0) = v$. Equipped with a retraction map, the basic form of the Riemannian optimization algorithms is depicted in Algo. 1, and there exist numerous variations according to which retraction map $\text{rt}_x(v)$ is utilized and how k -th step $v_k \in T_{x_k} \mathcal{M}$ is computed at each iteration [11], [20], [21]. In the next section, we discuss how the collision detection problem can be reformulated into a Riemannian optimization problem.

III. MTD PROBLEM AS RIEMANNIAN OPTIMIZATION

In this section, we introduce the reformulation of MTD problem into an unconstrained optimization on \mathbb{S}^{n-1} , which is the space of unit normal vectors of the boundary of a strictly convex body in \mathbb{R}^n , by observing the peculiar relation between the minimum distance problem and the support function. Then, we exhibit that the problem with convex bodies of class C^2 can be solved by RTR method ensuring superlinear convergence rate as long as the penetration depth between the convex bodies is small enough.

A. Reformulating MTD Problem

We first introduce well known dual formulation of the minimum distance problem from [22]. Given convex sets $\mathcal{C}_1, \mathcal{C}_2$ with the corresponding support functions $h_{\mathcal{C}_1}, h_{\mathcal{C}_2}$, the minimum distance between the two sets are given by

$$\begin{aligned} \min_{p_1 \in \mathcal{C}_1, p_2 \in \mathcal{C}_2} \|p_1 - p_2\| &= - \min_{\|x\| \leq 1} (h_{\mathcal{C}_2}(x) + h_{\mathcal{C}_1}(-x)) \\ &= - \min_{\|x\| \leq 1} h_{\Delta \mathcal{C}}(x) \end{aligned}$$

with $\Delta \mathcal{C} = \mathcal{C}_2 - \mathcal{C}_1$, where the last equality holds by (3). Then, we can observe that if $0 \notin \Delta \mathcal{C}$, the minimum of $h_{\Delta \mathcal{C}}(x)$ is achieved only if $\|x\| = 1$ by the positive homogeneity property (2) of the support function. Noticing that $0 \notin \Delta \mathcal{C}$ is equivalent to $\mathcal{C}_1 \cap \mathcal{C}_2 = \emptyset$, it can be inferred that $\text{MTD}(\mathcal{C}_1, \mathcal{C}_2)$ coincides to the negative of the minimum of $h_{\Delta \mathcal{C}}|_{\mathbb{S}^{n-1}}$ whenever \mathcal{C}_1 and \mathcal{C}_2 are non-intersecting.

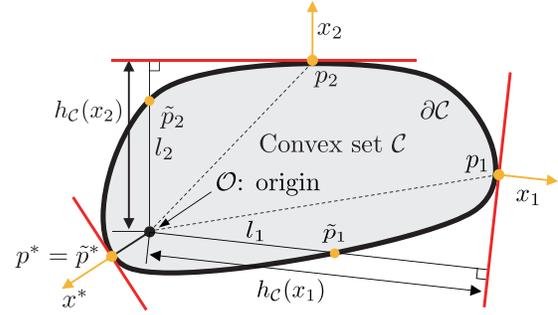


Fig. 3. For a compact convex set \mathcal{C} , points $\tilde{p}, p \in \partial \mathcal{C}$ determined by a normal vector $x \in \mathbb{S}^{n-1}$ satisfy the inequality $\|\tilde{p}\| \leq h_{\mathcal{C}}(x) \leq \|p\|$ and the equality holds if p is the closest point of the origin \mathcal{O} in the boundary $\partial \mathcal{C}$.

Moreover, the observation above can be extended to the case when \mathcal{C}_1 and \mathcal{C}_2 have intersection. First, consider an arbitrary compact convex set \mathcal{C} containing 0 inside and let $h_{\mathcal{C}}$ be its support function. Then, for any supporting hyperplane \mathcal{H} at each $p \in \partial \mathcal{C}$, the distance between \mathcal{H} and the origin is always less than or equal to $\|p\|$. Also, by the definition, $h_{\mathcal{C}}(x)$ for $x \in \mathbb{S}^{n-1}$ is a distance between a supporting hyperplane whose normal is x , hence we get

$$\min_{x \in \mathbb{S}^{n-1}} h_{\mathcal{C}}(x) \leq \min_{p \in \partial \mathcal{C}} \|p\|. \quad (5)$$

On the other side, consider a line segment l connecting the origin to its projection onto the supporting hyperplane \mathcal{H} of \mathcal{C} whose normal is $x \in \mathbb{S}^{n-1}$. Then, an intersection point \tilde{p} of l and $\partial \mathcal{C}$ between the origin and the supporting hyperplane \mathcal{H} satisfies

$$\|\tilde{p}\| \leq \text{length}(l) = h_{\mathcal{C}}(x) \quad (6)$$

since \mathcal{C} is convex as visualized in Fig. 3, and this directly induces

$$\min_{p \in \partial \mathcal{C}} \|p\| \leq \min_{x \in \mathbb{S}^{n-1}} h_{\mathcal{C}}(x). \quad (7)$$

Combining (5) and (7), we get the equality

$$\min_{p \in \partial \mathcal{C}} \|p\| = \min_{x \in \mathbb{S}^{n-1}} h_{\mathcal{C}}(x).$$

Now, recalling the statement in Remark 1, the minimum translational distance between two compact convex sets \mathcal{C}_1 and \mathcal{C}_2 is equivalent to the following:

$$\text{MTD}(\mathcal{C}_1, \mathcal{C}_2) = - \min_{x \in \mathbb{S}^{n-1}} h_{\Delta \mathcal{C}}(x). \quad (8)$$

Note that the optimal solution of (8) becomes the direction of separation with minimum translational distance.

B. MTD Problem as Riemannian Optimization

By (8), we can formulate $\text{MTD}(\mathcal{C}_1, \mathcal{C}_2)$ for two convex sets $\mathcal{C}_1, \mathcal{C}_2 \subset \mathbb{R}^n$ as a single unconstrained optimization problem on \mathbb{S}^{n-1} which is a compact manifold. This formulation implies that various tools from the field of Riemannian optimization are exploitable. Among the tools of Riemannian optimization, we utilize the Riemannian trust region (RTR) method [11] to solve $\text{MTD}(\mathcal{C}_1, \mathcal{C}_2)$ since our main focus is on smooth convex bodies allowing us to leverage second order information of the corresponding support functions.

Algorithm 2 Solving MTD problem

Given $\mathcal{C}_1, \mathcal{C}_2$ with support functions $h_{\mathcal{C}_1}, h_{\mathcal{C}_2}$
 Define $h_{\Delta\mathcal{C}}(x) := h_{\mathcal{C}_2}(x) + h_{\mathcal{C}_1}(-x)$
 Set maximum radius $\bar{d} > 0$, threshold $\rho' \in (0, 1/4)$ and
 optimality constants $\epsilon_g, \epsilon_H > 0$
 Initialize $x_0 \in \mathbb{S}^{n-1}, d_0 \in (0, \bar{d}), k = 0$

repeat

 Compute v_k solving (9) for x_k, d_k via tCG method

 Compute $x_k^+ = (x_k + v_k) / \|x_k + v_k\|$

 Compute $\rho_k = \frac{h_{\Delta\mathcal{C}}(x_k) - h_{\Delta\mathcal{C}}(x_k^+)}{m_{x_k}(0) - m_{x_k}(v_k)}$.

 Update next point x_{k+1} :

$$x_{k+1} = \begin{cases} x_k^+ & \text{if } \rho_k > \rho' \\ x_k & \text{otherwise} \end{cases}$$

 Update trust region radius d_{k+1}

$$d_{k+1} = \begin{cases} d_k/4 & \text{if } \rho_k < 1/4 \\ \min(2d_k, \bar{d}) & \text{if } \rho_k > 3/4 \\ d_k & \text{otherwise} \end{cases}$$

$k \leftarrow k + 1$

until (13)

RTR method can be seen as an analogy of the well known trust region method [23], as discussed in [11].

To begin with, let \mathbb{S}^{n-1} be endowed with a Riemannian metric $\langle \cdot, \cdot \rangle$ induced from \mathbb{R}^n , and $\text{rt}_x(v) := \frac{x+v}{\|x+v\|}$ be a retraction on \mathbb{S}^{n-1} which is clearly smooth. Also, assume that \mathcal{C}_1 and \mathcal{C}_2 are of class C^2 (i.e. support functions $h_{\mathcal{C}_1}$ and $h_{\mathcal{C}_2}$ are twice continuously differentiable on $\mathbb{R}^n \setminus \{0\}$). Given a k -th iteration point $x_k \in \mathbb{S}^{n-1}$ and a trust region radius $d_k > 0$, our algorithm with RTR method solves the following subproblem

$$\min_{v \in T_{x_k} \mathbb{S}^{n-1}} m_{x_k}(v) \text{ s.t. } \langle v, v \rangle_{x_k} \leq d_k^2, \quad (9)$$

for a model function $m_x : T_x \mathbb{S}^{n-1} \rightarrow \mathbb{R}$, which is a quadratic approximation of $h_{\Delta\mathcal{C}}$ on $T_x \mathbb{S}^{n-1}$ at $x \in \mathbb{S}^{n-1}$, as defined by

$$m_x(v) = h_{\Delta\mathcal{C}}(x) + \langle \text{grad } h_{\Delta\mathcal{C}}(x), v \rangle_x + \frac{1}{2} \langle v, \text{Hess } h_{\Delta\mathcal{C}}(x)v \rangle_x. \quad (10)$$

In (10), $\text{grad } h_{\Delta\mathcal{C}}(x)$ and $\text{Hess } h_{\Delta\mathcal{C}}(x)$ denote the Riemannian gradient and the hessian of $h_{\Delta\mathcal{C}}$ respectively. As shown in [19], the Riemannian gradient and hessian of an arbitrary smooth function $f : \mathbb{S}^{n-1} \rightarrow \mathbb{R}$ on a unit $(n-1)$ -sphere \mathbb{S}^{n-1} can be easily calculated by

$$\text{grad } f(x) = (I_n - xx^T) \frac{\partial f}{\partial x} \quad (11)$$

$$\text{Hess } f(x) = \frac{\partial^2 f}{\partial x^2} - \left(x^T \frac{\partial f}{\partial x} \right) I_n \quad (12)$$

where $\frac{\partial f}{\partial x} \in \mathbb{R}^n$ and $\frac{\partial^2 f}{\partial x^2} \in \mathbb{R}^{n \times n}$ are the regular gradient and the hessian of f on \mathbb{R}^n , and I_n is the $n \times n$ identity matrix.

In practice, the subproblem (9) is approximately solved since exactly solving it may be time consuming. We selected the truncated conjugate gradient (tCG) method [24] to approximately solve (9), which is commonly adopted for RTR method [11], [25], [19]. After solving the subproblem, we update x_{k+1} with $\text{rt}_{x_k}(v_k)$ according to the expected improvement of the function value where $v_k \in T_{x_k} \mathbb{S}^{n-1}$ is the k -th step resulted from tCG method. The procedure above repeats until the following termination criterion is satisfied for $x = x_k$:

$$\| \text{grad } h_{\Delta\mathcal{C}}(x) \| < \epsilon_g, \Lambda_{\min, x}(\text{Hess } h_{\Delta\mathcal{C}}(x)) > -\epsilon_H \quad (13)$$

where $\epsilon_g > 0$ and $\epsilon_H > 0$ are sufficiently small optimality constants, and $\Lambda_{\min, x}(H)$ for an $n \times n$ matrix H denotes the minimum eigenvalue of H on $T_x \mathbb{S}^{n-1}$. The summarized procedure of solving the MTD problem based on RTR method is given in from Algo. 2.

Note that we utilize representations of $\text{grad } h_{\Delta\mathcal{C}}(x)$ and $\text{Hess } h_{\Delta\mathcal{C}}(x)$ projected to $T_x \mathbb{S}^{n-1}$ when we solve (9) and compute (13). The projected representations are computed by $V^T \text{grad } h_{\Delta\mathcal{C}}(x) \in \mathbb{R}^{n-1}$ and $V^T \text{Hess } h_{\Delta\mathcal{C}}(x)V \in \mathbb{R}^{(n-1) \times (n-1)}$ with an arbitrary $n \times (n-1)$ matrix V which column vectors form an orthonormal basis on $T_x \mathbb{S}^{n-1}$ (i.e. $V^T x = 0, V^T V = I_{n-1}$). We also mention that the representation (12) is actually not mathematically well-defined since the precise definition of the Riemannian hessian on \mathbb{S}^{n-1} at $x \in \mathbb{S}^{n-1}$ is restricted to tangent space $T_x \mathbb{S}^{n-1}$. Nevertheless, the representation (12) does not hinder the validity of subsequent arguments presented in the paper.

C. Convergence of MTD Problem with RTR Method

It is shown in [11] that RTR method ensures local convergence if an objective function $f : \mathcal{M} \rightarrow \mathbb{R}$ is a C^2 function and \mathcal{M} is a compact manifold. Additionally, if $x^* \in \mathcal{M}$ is a non-degenerate local minimizer (i.e. $\text{grad } f(x^*) = 0$ and $\text{Hess } f(x^*)$ is positive definite on $T_{x^*} \mathbb{S}^{n-1}$), then RTR method exhibits superlinear convergence rate around a neighborhood of x^* [11]. It is noticed that our setting of collision detection satisfies the first statement since \mathbb{S}^{n-1} is compact and we are dealing with convex bodies of class C^2 . Moreover, we show that the the global minimizer satisfies non-degeneracy as long as the penetration depth is small enough. To concretely state the condition for the non-degeneracy, we define the minimum radius of curvature $\rho_{\min}(\mathcal{C})$ of a convex set \mathcal{C} of class C^2 by

$$\rho_{\min}(\mathcal{C}) = \min_{x \in \mathbb{S}^{n-1}} \Lambda_{\min, x} \left(\frac{\partial^2 h_{\mathcal{C}}}{\partial x^2}(x) \right).$$

Then, the following theorem encapsulates our arguments about the convergence of Algo. 2.

Theorem 1: Let \mathcal{C}_1 and \mathcal{C}_2 be compact convex bodies of class C^2 with the associated support functions $h_{\mathcal{C}_1}, h_{\mathcal{C}_2}$. Then the following holds:

- 1) Algo. 2 always converges to a local minimizer of (8).
- 2) There exists a neighborhood around a global minimizer of (8) such that Algo. 2 exhibits superlinear convergence if $\text{MTD}(\mathcal{C}_1, \mathcal{C}_2) > -(\rho_{\min}(\mathcal{C}_1) + \rho_{\min}(\mathcal{C}_2))$.

Proof: As we have discussed in the beginning of Sec. III-C, the item 1) holds by the convergence property of RTR method [11] induced from the compactness of \mathbb{S}^{n-1} . Now, observe that the Riemannian hessian at x^* becomes

$$\begin{aligned} \text{Hess } h_{\Delta C}(x) &= \frac{\partial^2 h_{\Delta C}}{\partial x^2}(x) - h_{\Delta C}(x)I_n \\ &= \frac{\partial^2 h_{C_1}}{\partial x^2}(-x) + \frac{\partial^2 h_{C_2}}{\partial x^2}(x) - h_{\Delta C}(x)I_n \end{aligned}$$

by (12) and the definition of the support function. Since a global minimizer x^* satisfies $h_{\Delta C}(x^*) = -\text{MTD}(C_1, C_2)$, the above equation and the definition of $\rho_{\min}(\cdot)$ imply that $\text{Hess } h_{\Delta C}(x^*)$ is positive definite on $T_{x^*}\mathbb{S}^{n-1}$ and we get the result of the item 2). ■

Whether the algorithm always converges to a global minimizer cannot be ensured by Theorem 1. However, we empirically observed that Algo. 2 converges to the global minimizer in every non-penetration scenarios when x_0 is defined by

$$x_0 = \frac{p_1 - p_2}{\|p_1 - p_2\|} \quad (14)$$

with centroids p_1 and p_2 of C_1 and C_2 , respectively, which are often known. In addition, Theorem 1 implies that the convergence of Algo. 2 to a global minimizer always exhibits fast convergence up to a small penetration depth, which is commonly assumed for applications such as simulation. Moreover, by the following remark, the allowable penetration depth can be pre-defined, purely dependent to the shapes of the convex sets in the collision detection problem.

Remark 2: The minimum radius of curvature of a convex set is invariant under rigid-body transformations i.e.

$$\rho_{\min}(\bar{C}(R, t)) = \rho_{\min}(C), \quad \bar{C}(R, t) = \{Rp + t \mid p \in C\}$$

for any $R \in \text{SO}(n)$ and $t \in \mathbb{R}^n$.

The remark can be easily induced from the relation

$$h_{\bar{C}(R, t)}(x) = h_C(R^T x) + t \cdot x. \quad (15)$$

where $h_{\bar{C}(R, t)}$ and h_C are support functions of $\bar{C}(R, t)$ and C , respectively. Note that we call $(R, t) \in \text{SE}(n)$ as a configuration of $\bar{C}(R, t)$.

Up to now, we have proposed the algorithm solving (8) and its convergence property largely derived from our method's exploitation of second-order information. We now conclude the section presenting additional useful property of our method. Consider convex bodies $\bar{C}_1(R_1, t_1)$ and $\bar{C}_2(R_2, t_2)$ transformed from C_1 and C_2 with configurations $(R_1, t_1), (R_2, t_2) \in \text{SE}(n)$, respectively. Then, since the corresponding support functions are C^2 functions, the first optimality equation of (8) is totally differentiable with respect to the configurations. As a result, we can compute the derivatives of contact features from the total derivative of the first optimality equation by utilizing the implicit function theorem.

IV. IMPLEMENTATION

So far, we have discussed about the novel formulation for the MTD model to which any general C^2 support functions in \mathbb{R}^n are applicable. Although a large portion of geometric representations in simulation or planning is based on non-differentiable objects such as meshes or polytopes, differentiable convex representations can still be used for simulation or planning with fairly general geometric shapes by exploiting, for example, convex decomposition technique [26], [27]. In this section, we introduce differentiable geometric representations of convex sets in \mathbb{R}^3 suited to our MTD formulation.

A. Superquadrics

Superquadrics have been widely used for representing smooth primitive shapes such as superellipsoids [28], [29], [8]. A superquadric \mathcal{SQ} is an implicit surface in \mathbb{R}^3 defined as a zero level set of a function $f_{\mathcal{SQ}} : \mathbb{R}^3 \rightarrow \mathbb{R}$, (i.e. $\mathcal{SQ} = f_{\mathcal{SQ}}^{-1}(0)$), where f is defined by

$$f_{\mathcal{SQ}}(p) = \left(\left(\frac{p_1}{a} \right)^{\frac{2}{\alpha_2}} + \left(\frac{p_2}{b} \right)^{\frac{2}{\alpha_2}} \right)^{\frac{\alpha_2}{\alpha_1}} + \left(\frac{p_3}{c} \right)^{\frac{2}{\alpha_1}} - 1 \quad (16)$$

for $p = (p_1, p_2, p_3) \in \mathbb{R}^3$, scale factors $a, b, c > 0$ along each axis and shape related parameters $\alpha_1, \alpha_2 > 0$. It is known that \mathcal{SQ} is the boundary of a strictly convex shape when $\alpha_1, \alpha_2 \in (0, 2)$. As mentioned in [8], from (16), we can get a closed form parametrization (or an inverse Gauss map) $p_{\mathcal{SQ}}(x)$ of \mathcal{SQ} with a surface normal $x \in \mathbb{S}^2$. In addition, noting that $p_{\mathcal{SQ}}(\lambda x) = \lambda p_{\mathcal{SQ}}(x)$ for any $\lambda > 0$ and $x \in \mathbb{S}^2$, we can define a support function $h_{\mathcal{SQ}}$ associated with \mathcal{SQ} by

$$h_{\mathcal{SQ}}(x) = x \cdot p_{\mathcal{SQ}}(x)$$

which can be inferred from the relation between a surface point and the corresponding surface normal in (1). $h_{\mathcal{SQ}}(x)$ is clearly differentiable and a C^2 function when $\alpha_1 \geq 1$ and $\alpha_2 \geq 1$. On the other hand, $h_{\mathcal{SQ}}(x)$ is not a C^2 function when $\alpha_1 < 1$ or $\alpha_2 < 1$ due to the degeneracy of the standard hessian matrix of $f_{\mathcal{SQ}}$ in \mathbb{R}^3 at the certain points of $(\pm a, 0, 0)$, $(0, \pm b, 0)$, $(0, 0, \pm c)$. This degeneracy implies that the inverse Gauss map $p_{\mathcal{SQ}}(x)$ is non-differentiable at the surface normals corresponding to the points above. To ensure that $h_{\mathcal{SQ}}(x)$ is a C^2 function, we only deal with superquadrics with $\alpha_1, \alpha_2 \in [1, 2)$ when evaluating our method. Also, for the cases when $\alpha_1, \alpha_2 > 1$, we add the support function of a sphere with a small radius ($\sim 10^{-4}$) to the support function $h_{\mathcal{SQ}}(x)$ enhancing our method by imposing the positive definiteness to the hessian matrix of $h_{\mathcal{SQ}}(x)$.

B. Prescribed Support Functions

Suggested by Lee et. al. [9], a prescribed support function smoothly approximates a support function of a convex hull with finite vertices. Given a discrete convex hull \mathcal{C}_d with

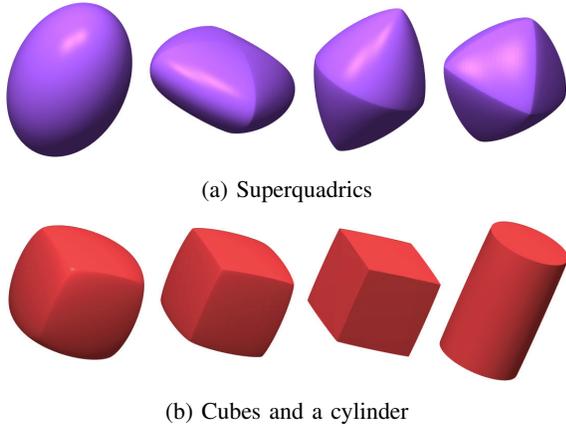


Fig. 4. (a) Superquadrics represented by the implicit function (16). Left to right ellipsoid, rectangular superellipsoid, double cone, double pyramid with parameters specified in Table I. (b) Cubes and a cylinder represented by the prescribed support function (17). Left to right the sharpness parameters are set by $p = 5, 10, 50, 40$.

vertices $\{p_1, \dots, p_n\} \subset \mathbb{R}^n$, the corresponding prescribed support function $h_C : \mathbb{R}^n \rightarrow \mathbb{R}$ is defined by

$$h_C(x) = \left(\sum_{i=1}^n \{\max(p_i \cdot x, 0)\}^\beta \right)^{1/\beta} \quad (17)$$

where $\beta > 2$ denotes the sharpness of the smoothed convex geometry \mathcal{C} which is uniquely determined by h_C as shown in [9]. We can observe that the smoothed geometry \mathcal{C} converges to the original discrete geometry \mathcal{C}_d as β increases. Moreover, the prescribed support function h_C is a C^2 function on $\mathbb{R}^n \setminus \{0\}$ and for any $x \in \mathbb{S}^{n-1}$,

$$v^T \frac{d^2 h_C}{dx^2}(x) v > 0, \quad \forall v \in T_x \mathbb{S}^{n-1} \quad (18)$$

if there exists at least 3 linearly independent vertices p_i such that $p_i \cdot x > 0$. In light of the result of Theorem 1, these properties make a prescribed support function preferable to be employed with our novel MTD method.

Another variant of a prescribed support function can represent solids of rotation in \mathbb{R}^3 . Consider a convex body of rotation \mathcal{C}_d generated by rotating a convex hull around z -axis, where the convex hull is defined on xz -plane with vertices $\{p_1, p_2, \dots, p_n\} \subset \mathbb{R}^2$ symmetric to z -axis. Then, the corresponding prescribed support function $h_C : \mathbb{R}^3 \rightarrow \mathbb{R}$ is defined by

$$h_C(x_1, x_2, x_3) = \left(\sum_{i=1}^n \{\max(p_i \cdot r, 0)\}^\beta \right)^{1/\beta} \quad (19)$$

where $r = (\sqrt{x_1^2 + x_2^2 + \gamma x_3^2}, x_3) \in \mathbb{R}^2$ and $\gamma > 0$ is a parameter determining flatness at the top and bottom of the convex body. As γ goes to zero and β increases, the geometry represented by h_C converges to a body generated by rotating the 2D discrete convex hull. Note that h_C also satisfies the property (18).

TABLE I
SHAPE PARAMETERS FOR PRIMITIVE SHAPES

shape	parameters	shape	parameters
	$(a, b, c, \alpha_1, \alpha_2)$		(γ, β)
ellipsoid	(0.5, 0.5, 0.7, 1.0, 1.0)	cube 1	$(\cdot, 5)$
superellipsoid	(0.7, 0.7, 0.35, 1.0, 1.5)	cube 2	$(\cdot, 10)$
double cone	(0.5, 0.5, 0.7, 1.5, 1.0)	cube 3	$(\cdot, 50)$
double pyramid	(0.6, 0.6, 0.6, 1.5, 1.5)	cylinder	$(10^{-3}, 40)$

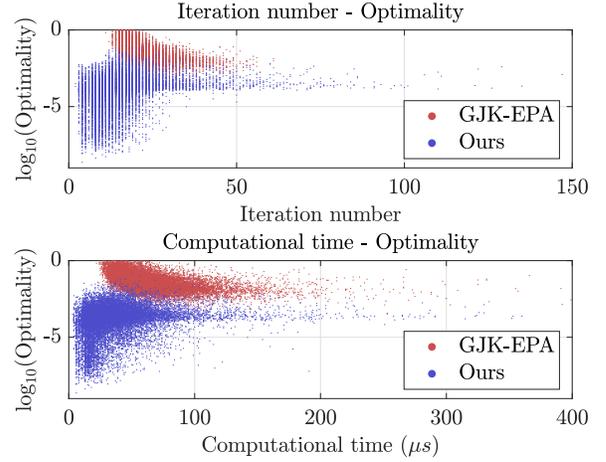


Fig. 5. The plots describe the relation between terminal optimality measure (20) with the number of iterations and computational time taken until termination for each method.

V. EVALUATION

In this section, we present the evaluation result of our MTD formulation in \mathbb{R}^3 , comparing to GJK-EPA method [1], [2] with respect to optimality, iteration numbers and computation time. To ensure that each algorithm solves the MTD problem with the same geometries, both methods were set to utilize the same support function for each shape. As a measure of optimality, we adopted the norm of a cross product between a normal direction x and a unit vector aligned to the line connecting two witness points:

$$\text{Optimality} = \frac{\|x \times (s_2(x) - s_1(-x))\|}{\|s_2(x) - s_1(-x)\|} \quad (20)$$

which measures how much the direction of the normal vector x coincides to the direction of the vector between witness points $s_1(x) \in \mathcal{C}_1$, $s_2(-x) \in \mathcal{C}_2$. The evaluation studies were implemented in C++ and performed in a PC with CPU Intel(R) Core(TM) i5-12400F 2.50 GHz. For GJK-EPA method, open source collision detection libraries *libccd* [30] and *FCL* [31] were adopted. Note that the evaluation primarily focuses for the cases when two convex bodies are in contact since the performance of EPA method, when penetration occurs, deteriorates with smooth convex bodies.

A. Termination Criteria

Our method employs the termination criterion defined by (13), whereas EPA method terminates when the difference between the distance from newly expanded supporting plane of a Minkowski sum to the origin and the previous nearest

TABLE II
AVERAGED COMPUTATIONAL TIME FOR THE MTD METHODS

GJK-EPA	SQ*				PSF*			
	ellip.	super ellip.	double cone	double pyramid	cube 1	cube 2	cube 3	cylinder
Ours (μs)	ellip.	super ellip.	double cone	double pyramid	cube 1	cube 2	cube 3	cylinder
ellipsoid	58	68	73	67	66	65	59	68
	14	30	37	31	25	22	24	56
super ellipsoid	-	72	77	67	65	63	53	68
	-	52	61	43	27	29	37	57
double cone	-	-	69	67	78	71	64	79
	-	-	56	50	35	38	45	72
double pyramid	-	-	-	68	67	59	55	71
	-	-	-	36	25	26	33	47
cube 1	-	-	-	-	67	65	54	68
	-	-	-	-	12	15	22	32
cube 2	-	-	-	-	-	55	51	68
	-	-	-	-	-	15	22	35
cube 3	-	-	-	-	-	-	50	67
	-	-	-	-	-	-	25	42
cylinder	-	-	-	-	-	-	-	83
	-	-	-	-	-	-	-	60

* SQ: superquadric, PSF: prescribed support function

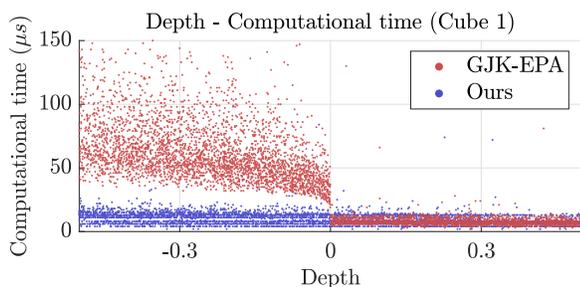


Fig. 6. The plot presents a comparison of the relationship between penetration depth and computational for collision detection of a pair of cube 1 - cube 1.

distance falls below predefined thresholds, $\epsilon_{EPA} > 0$. These different termination criteria result in differing values of the optimality measure upon termination of Algo. 2 process complicating the adjustment of ϵ_g in (13) and ϵ_{EPA} to achieve consistent levels of optimality. Nonetheless, we set $\epsilon_g = \epsilon_{EPA} = 10^{-4}$ yielding the level of the difference between resulted penetration depths to be $10^{-5} \sim 10^{-3}$ which is affordable in many robotic applications.

B. Evaluation and Analysis

The performance evaluation was conducted using various primitive shapes, including an ellipsoid, a rectangular superellipsoid, a double cone, a double pyramid, cubes with varying sharpnesses, and a cylinder represented by superquadrics and prescribed support functions. Specific parameters for these primitive shapes are detailed in Table I, and visualized images are provided in Fig. 4. For each pair of the primitives, 500 random poses are generated and collision detections were performed respectively using both our method and GJK-EPA method. The results are depicted in Fig. 5, illustrating the distribution of the optimality measures against iteration numbers and computational times. The averaged computational times for each primitive pair, with 10,000 random poses, are summarized in Table II. Note that the numbers in Table II are rounded to the closest integer.



Fig. 7. Overlapped visualizations of the selected YCB objects (textured) and the corresponding fitted models (transparent) with prescribed support functions. In the figure, the fitted models are scaled larger than the original ones for clear visualizations.

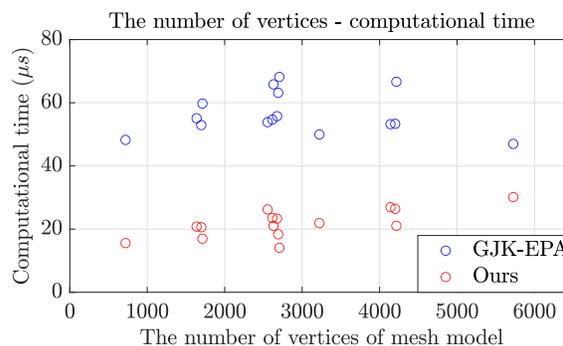


Fig. 8. The plot illustrates the averaged computational times of GJK-EPA method and ours against the sum of the vertices' number of each YCB mesh model pair.

Our method demonstrates superior performance over GJK-EPA method, exhibiting better optimality measures and smaller iteration numbers. The superiority stems from our method's exploitation of second-order information of the given geometries, in contrast to the derivative-free nature of GJK-EPA. Moreover, as depicted in Fig. 6, our method demonstrates consistent computational time across varying penetration depths, which is attributed to the convergence rate guaranteed by the well-established Riemannian optimization theory.

Lastly, we note that the resulting depth $MTD^+(C_1, C_2)$ obtained by our method is smaller than those obtained by GJK-EPA method in the 99.96% of the evaluation cases. This shows that our method converges to a global minimizer in almost every cases in practice.

C. Demonstration on YCB dataset

In addition to the primitive shapes, we demonstrate the significant enhancement of our method in collision detection with more complex shapes compared to GJK-EPA method. For this, we selected 5 real world models from YCB dataset [10] consisting of smooth surfaces and computed their convex hulls as our methods and GJK-EPA work on convex shapes. For GJK-EPA method, we adopted hill-climbing method [5] implemented by *FCL* [31] library to compute the support function efficiently. In addition, we fitted each

convex hull to a shape represented by prescribed support functions (17) with 20 vertices by solving simple fitting optimization problems. The selected YCB objects and the fitted geometries are illustrated in Fig. 7. We conducted our method with the fitted prescribed functions and GJK-EPA with the discrete convex hulls of the selected YCB mesh models for 10,000 times for each combination of the objects, and recorded the averaged computational times.

The result is depicted Fig. 8 where the log scaled computational time is plotted against the sum of the vertices' number of each YCB mesh model pair. It is noteworthy to emphasize that utilizing our method with the fitted support functions for the scenarios involving smooth surfaces can significantly enhance the speed of collision detection.

VI. CONCLUSION

This paper is the first to establish the collision detection problem between smooth convex bodies as a Riemannian optimization problem. Employing RTR method, our collision detection algorithm always converges to a local minimizer and exhibits superlinear convergence around a global minimizer within a specific small penetration depth. We also introduce classes of support functions applicable to our collision detection algorithm. By exploiting second-order features of given geometries, our algorithm outperforms conventional GJK-EPA algorithm in most of the scenarios involving penetration of primitive shapes. Notably, the demonstration results with YCB data shows our algorithm's potential to reduce computational load of collision detections between smooth surfaces represented by fine mesh-based models which has been a longstanding challenge for GJK-EPA based methods. Although collision-free scenarios are not extensively demonstrated in this paper, we also mention that our method exhibits performance comparable to GJK method in such cases.

For future works, we identify two key directions. Firstly, we aim to develop methods for designing and fitting prescribed support functions to known mesh-based models, recognizing their importance for our algorithm's performance from the demonstration results. Secondly, we intend to explore our method's effectiveness in applications such as simulation and planning leveraging resultant contact features.

REFERENCES

- [1] E. G. Gilbert, D. W. Johnson, and S. K. Selvaraj. A fast procedure for computing the distance between complex objects in three-dimensional space. *IEEE Journal on Robotics and Automation*, 4(2):193–203, 1988.
- [2] G. V. D. Bergen. Proximity queries and penetration depth computation on 3d game objects. In *Game developers conference*, 2001.
- [3] Ralph T. R. *Convex Analysis*. Princeton University Press, Princeton, 1970.
- [4] E. G. Gilbert and C. P. Foo. Computing the distance between general convex objects in three-dimensional space. *IEEE Transactions on Robotics and Automation*, 6(1):53–61, 1990.
- [5] S. Cameron. Enhancing gjk: Computing minimum and penetration distances between convex polyhedra. In *Proceedings of International Conference on Robotics and Automation (ICRA)*, volume 4, pages 3112–3117. IEEE, 1997.
- [6] C. J. Ong and E. G. Gilbert. Fast versions of the gilbert-johnson-keerthi distance algorithm: additional results and comparisons. *IEEE Transactions on Robotics and Automation*, 17(4):531–539, 2001.
- [7] L. Montaut, Q. Le Lidec, V. Petrik, J. Sivic, and J. Carpentier. Collision detection accelerated: An optimization perspective. In *RSS 2022-Robotics: Science and Systems*, 2022.
- [8] S. Ruan, X. Wang, and G. S. Chirikjian. Collision detection for unions of convex bodies with smooth boundaries using closed-form contact space parameterization. *IEEE Robotics and Automation Letters*, 7(4):9485–9492, 2022.
- [9] J. Lee, M. Lee, and D. Lee. Uncertain pose estimation during contact tasks using differentiable contact features. *RSS 2023-Robotics: Science and Systems*, 2023.
- [10] B. Calli, A. Singh, A. Walsman, S. Srinivasa, P. Abbeel, and A. M. Dollar. The ycb object and model set: Towards common benchmarks for manipulation research. In *2015 International Conference on Advanced Robotics (ICAR)*, pages 510–517, 2015.
- [11] P. A. Absil, C. G. Baker, and K. A. Gallivan. Trust-region methods on riemannian manifolds. *Foundations of Computational Mathematics*, 7:303–330, 2007.
- [12] M. Lee, J. Lee, and D. Lee. Differentiable dynamics simulation using invariant contact mapping and damped contact force. In *2023 IEEE International Conference on Robotics and Automation (ICRA)*, pages 11683–11689. IEEE, 2023.
- [13] X. Zhang, A. Liniger, and F. Borrelli. Optimization-based collision avoidance. *IEEE Transactions on Control Systems Technology*, 29(3):972–983, 2020.
- [14] J. Choe, J. Lee, H. Yang, H. N. Nguyen, and D. Lee. Sequential trajectory optimization for externally-actuated modular manipulators with joint locking. In *2024 IEEE International Conference on Robotics and Automation (ICRA)*, pages 8700–8706. IEEE, 2024.
- [15] R. Schneider. *Convex bodies: the Brunn–Minkowski theory*. Number 151. Cambridge University Press, 2014.
- [16] M. P. Do Carmo. *Differential geometry of curves and surfaces: revised and updated second edition*. Courier Dover Publications, 2016.
- [17] J. M. Lee. *Introduction to Riemannian manifolds*, volume 2. Springer, 2018.
- [18] S. Cameron and R. K. Culley. Determining the minimum translational distance between two convex polyhedra. In *Proceedings of International Conference on Robotics and Automation (ICRA)*, volume 3, pages 591–596. IEEE, 1986.
- [19] N. Boumal. *An introduction to optimization on smooth manifolds*. Cambridge University Press, 2023.
- [20] C. Liu and N. Boumal. Simple algorithms for optimization on riemannian manifolds with constraints. *Applied Mathematics & Optimization*, 82:949–981, 2020.
- [21] M. Weber and S. Sra. Riemannian optimization via frank-wolfe methods. *Mathematical Programming*, 199(1-2):525–556, 2023.
- [22] S. P. Boyd and L. Vandenberghe. *Convex Optimization*. Cambridge University Press, 2004.
- [23] A. R. Conn, N. I. Gould, and P. L. Toint. *Trust region methods*. SIAM, 2000.
- [24] T. Steihaug. The conjugate gradient method and trust regions in large scale optimization. *SIAM Journal on Numerical Analysis*, 20(3):626–637, 1983.
- [25] Y. Yuan. Recent advances in trust region algorithms. *Mathematical Programming*, 151:249–281, 2015.
- [26] J. Lien and N. M. Amato. Approximate convex decomposition of polyhedra. In *Proceedings of the 2007 ACM symposium on Solid and physical modeling*, pages 121–131, 2007.
- [27] B. Deng, K. Genova, S. Yazdani, S. Bouaziz, G. Hinton, and A. Tagliasacchi. Cvxnet: Learnable convex decomposition. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 31–44, 2020.
- [28] A. H. Barr. Superquadrics and angle-preserving transformations. *IEEE Computer Graphics and Applications*, 1(1):11–23, 1981.
- [29] D. Paschalidou, A. O. Ulusoy, and A. Geiger. Superquadrics revisited: Learning 3d shape parsing beyond cuboids. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10344–10353, 2019.
- [30] D. Fiser et. al. libccd: Library for collision detection between convex shapes. url:<https://github.com/danfis/libccd>, 2018.
- [31] J. Pan, S. Chitta, and D. Manocha. Fcl: A general purpose library for collision and proximity queries. In *2012 IEEE International Conference on Robotics and Automation (ICRA)*, pages 3859–3866. IEEE, 2012.