IEEE Robotics and Automation Letters (RA-L) paper, presented at ICRA 2026, Vienna, Austria. Cite as RA-L paper.

Enhanced Probabilistic Collision Detection for Motion Planning Under Sensing Uncertainty

Xiaoli Wang[†1], Sipu Ruan[†1], Xin Meng[1], and Gregory S. Chirikjian[*1,2]

***Abstract*—Probabilistic collision detection (PCD) is essential in motion planning for robots operating in unstructured environments, where considering sensing uncertainty helps prevent damage. Existing PCD methods mainly use simplified geometric models and address only position estimation errors. This paper presents an enhanced PCD method with two key advancements: (a) using superquadrics for more accurate shape approximation and (b) accounting for both position and orientation estimation errors to improve robustness under sensing uncertainty. Our method first computes an enlarged surface for each object that encapsulates its observed rotated copies, thereby addressing the orientation estimation errors. Then, the collision probability is formulated as a chance-constraint problem that is solved with a tight upper bound. Both steps leverage the recently developed closed-form normal parameterized surface expression of superquadrics. Results show that our PCD method is twice as close to the Monte-Carlo sampled baseline as the best existing PCD method and reduces path length by 30% and planning time by 37%, respectively. A Real2Sim2Real pipeline further validates the importance of considering orientation estimation errors, showing that the collision probability of executing the planned path is only 2%, compared to 9% and 29% when considering only position estimation errors or no errors at all.**

***Index Terms*—Collision avoidance, planning under uncertainty, motion and path planning**

I. Introduction

COLLISION detection is essential to motion planning, which helps to prevent robots from colliding with their surroundings. Although traditional collision detection methods have been developed for decades, they usually assume perfect knowledge of the states of robots and environments [1]. This assumption does not apply in most real-world applications, especially for service robots with a high degree of freedom (DOF) manipulating objects in domestic settings. In such cases, robots will need to interact with objects closely, but the pose estimation of objects is often affected by occlusions and sensor inaccuracies. The sensing uncertainty may cause unexpected collisions that lead to a failure of manipulation and damage the robot and the surroundings. An example is shown in Fig. 1 (c-d).

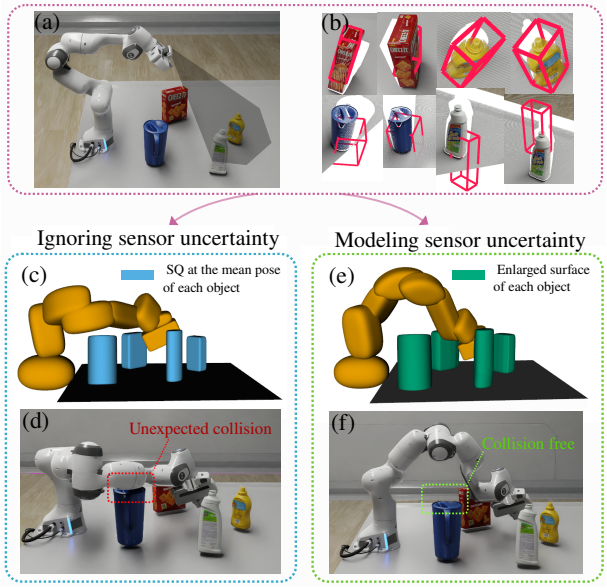


Fig. 1. Comparison of motion planning results with and without taking into account sensing uncertainty. (a-b) Pose estimates of each object using an existing method in Isaac Sim [2]. (c) Superquadric representation of each robot link at its ground truth pose (yellow) and of each object at its estimated mean pose (blue), where the robot configuration is justified as valid by a deterministic collision checker [1]. (d) Snapshot of a collision between the robot and an object due to unmodeled pose uncertainty. (e) The enlarged surface that encapsulates many rotated copies of the objects due to orientation errors in pose estimates (green), and the robot configuration that is justified as valid by our probabilistic collision checker. (f) Corresponding snapshot of the robot showing no collision with objects.

Compared with deterministic collision detection, probabilistic collision detection (PCD) takes into account the sensing uncertainty and calculates the collision probability between two bodies in a single query. When a PCD is used as the collision checker in a motion planner for a robot, it gives the collision probability between each link of the robot and each environmental object. The PCD outcome enables the motion planner to quantify the level of safety, ensuring that the overall collision probability between the robot and objects along the planned path is lower than a given threshold.

However, existing PCD methods usually use a simplified geometric model (e.g., points or ellipsoids) for the robot and environmental objects and only account for position estimation errors [3]–[6]. These assumptions may lead to less robustness when complex environmental objects cannot be accurately represented by simple geometries and when their orientation

Manuscript received: February, 18, 2025; Revised July, 20, 2025; Accepted August, 10, 2025. This paper was recommended for publication by Editor O. Stasse upon evaluation of the Associate Editor and Reviewers' comments. This work was supported by NUS Startup grants A-0009059-02-00, A-0009059-03-00, CDE Board account E-465-00-0009-01, and National Research Foundation, Singapore, under its Medium Sized Centre Programme - Centre for Advanced Robotics Technology Innovation (CARTIN), subaward A-0009428-08-00.

[1]Xiaoli Wang, Sipu Ruan, Xin Meng, and Gregory S. Chirikjian are with Department of Mechanical Engineering, National University of Singapore, Singapore. (email: wangxiaoli@u.nus.edu, rsp01@163.com, mengxin@u.nus.edu, mpegre@nus.edu.sg)

[2]Gregory S. Chirikjian is also with Department of Mechanical Engineering, University of Delaware, USA. (email: gchirik@udel.edu)

[†]The authors contribute equally. [*]Corresponding author.

Digital Object Identifier (DOI): see top of this page.

IEEE Robotics and Automation Letters (RA-L) paper, presented at ICRA 2026, Vienna, Austria. Cite as RA-L paper.

estimation errors cannot be ignored.

To address these limitations, this paper presents an enhanced PCD method with two key advancements. First, it supports using a broad class of geometric models, specifically superquadrics, to approximate the true shape of environmental objects accurately. Superquadrics are a family of geometric shapes defined by five parameters resembling ellipsoids and other quadrics, as introduced in Section III-D. This flexibility allows them to accurately approximate the standard collision geometric models (e.g., bounding box or cylinder) or represent details of complex objects. An example of the superquadrics representation for each robot link and environmental object is shown in Fig. 1 (c). We applied an existing method for the superquadric-based shape fitting [7], and compared the relative goodness of model fit using superquadrics and ellipsoids in the Section V-A. The second key advancement is that the proposed method accounts for object position and orientation estimation errors, increasing robustness against sensing uncertainty. More specifically, we proposed to use an enlarged surface that encapsulates many rotated copies of the objects due to orientation errors in pose estimates, as shown in Fig. 1 (e). Overall, these improvements enhance the reliability of PCD in unstructured environments.

We conducted a benchmark on PCD methods and found that our method considerably refines the accuracy of collision probability estimation. In addition, we examined how the accuracy and computation time of PCD methods influence the performance of motion planners. Results show that with our PCD method, planners find more efficient paths within less planning time, especially in cluttered scenes. Furthermore, we designed a Real2Sim2Real pipeline to assess the necessity of considering the orientation estimation errors of objects. We compared three different kinds of planners: 1) deterministic; 2) PCD including position errors; 3) PCD with both position and orientation errors. Results show that when both the position and orientation uncertainties are considered in motion planning, the collision probability during execution consistently exhibits the lowest risk.

The major contributions of this work are:

- Whereas previous methods only handle positional uncertainty, we present a novel method to handle orientation uncertainty using an enlarged parametric surface that encapsulates rotated copies of the object.
- Tight approximation for collision probability: An efficient linear chance constraint is proposed that gives a tight upper bound for collision probability under position errors.
- Real2Sim2Real evaluation pipeline: A new pipeline that quantifies the reliability of planned paths by simulating their execution under real-world sensing uncertainties.

II. Related Work

Recently, PCD has been developed to account for uncertainties in robot controllers and environment sensing to ensure the safety of robots operating in the real world, such as drones and autonomous cars [3] [8]. However, PCD for high-DOF robots is particularly challenging as they often need to interact closely with objects in cluttered scenes, such as during pick-and-place tasks, so even small errors in depth or pose estimation can add up and raise the chance of undetected collisions.

Early PCD methods, such as Monte Carlo-based approaches, compute the collision probability accurately but are computationally demanding, limiting their practical use in real-time applications [9]. Some PCD methods that have closed-form solutions are fast to compute and are suitable for high-speed robots [3], [10], [11]. Nevertheless, they only support using simple geometric primitives (e.g., points, spheres, or ellipsoids) to represent robots and environmental objects, which can give a conservative result if the objects cannot be accurately represented. Moreover, they only consider the position estimation errors of objects. Although some PCD methods support convex complex geometric models (e.g., mesh), their performance either depends on the surface complexity of the model [6] or needs to be iteratively improved [8]. Nevertheless, they only support the position estimation errors of objects. Learning-based methods use the point cloud of the environment and do not assume a probabilistic model per object, but the computation time is too expensive, and retraining is required for new robots. [12].

Despite these advancements, current PCD methods still face limitations in accurately modeling complex shapes and handling combined position and orientation uncertainties. This work addresses these gaps by introducing a robust PCD method that leverages superquadrics for improved shape approximation and integrates position and orientation uncertainties for enhanced robustness in real-world applications.

III. Preliminary

A. Probabilistic collision detection

The collision condition for two convex bodies S_1, S_2 with exact poses can be written as

$$\text{if } S_1 \cap S_2 \neq \varnothing \Leftrightarrow \mathbf{p}_2 \in S_1 \oplus (-S_2)\,.$$

where $S_1 \oplus (-S_2)$ is the Minkowski sum and $-S_2$ is the reflection of S_2 about its center $\mathbf{p}_2$.

If S_i is moved by $g_i = (R_i, \mathbf{t}_i) \in \mathrm{SE}(3)$, we denote the resulting body as $S_i^{g_i} = R_i S_i + \mathbf{t}_i$. If the poses g_1 and g_2 are inaccurate, the collision status is probabilistic and can be written as $\mathrm{P}(S_1^{g_1}, S_2^{g_2}) = P(S_1^{g_1} \cap S_2^{g_2} \neq \varnothing)$.

B. Collision probability under position sensing uncertainty

Previous PCD methods only consider inaccurate translation estimates $\mathbf{t}_i$ of objects. The inaccurate point set is $S_i^{\mathbf{t}_i} = S_i + \mathbf{t}_i$, and the collision condition becomes:

$$\begin{aligned}&(S_1 + \mathbf{t}_1) \cap (S_2 + \mathbf{t}_2) \neq \emptyset \\ &\Rightarrow \exists \mathbf{s}_1 \in S_1, \mathbf{s}_2 \in S_2 : (\mathbf{s}_2 + \mathbf{t}_2) - (\mathbf{s}_1 + \mathbf{t}_1) = \mathbf{0}\,, \\ &\Leftrightarrow \mathbf{t}_2 - \mathbf{t}_1 = \mathbf{s}_1 - \mathbf{s}_2, \Leftrightarrow \mathbf{t}_2 - \mathbf{t}_1 \in S_1 \oplus (-S_2).\end{aligned} \tag{1}$$

The collision probability $\mathrm{P}(S_1^{\mathbf{t}_1}, S_2^{\mathbf{t}_2})$ can be written as the integral:

$$\int_{\mathbb{R}^3}\int_{\mathbb{R}^3} \iota(\mathbf{t}_2 - \mathbf{t}_1 \in S_1 \oplus (-S_2))\rho_1(\mathbf{t}_1)\rho_2(\mathbf{t}_2) d\mathbf{t}_1 d\mathbf{t}_2\,, \tag{2}$$

IEEE Robotics and Automation Letters (RA-L) paper, presented at ICRA 2026, Vienna, Austria. Cite as RA-L paper.

where $\iota(\cdot)$ is an indicator function that returns 1 when the input condition is true and 0 otherwise. $\rho(\cdot)$ is the shorthand for the probability density function (pdf) defined by the argument.

The position error is usually modeled as zero-mean Gaussian distributed, i.e.,$\mathbf{t}_i \sim \mathcal{N}(\mathbf{0}, \Sigma_i)$, where Σ is the covariance. To make the formulas in this work concise and tight, we move the objects S_1 and S_2 to the global origin, and then $S_1 \oplus (-S_2)$ is centered at the global origin. Thus, the mean of the zero-mean additive position error is $\mathbf{t}_i \sim \mathcal{N}(\mathbf{p}_i, \Sigma_i)$, where $\mathbf{p}_i$ is the center position of object S_i.

C. Linear chance constraint approximation

The collision probability in (2) has no closed-form solution, and its numerical solution is computationally expensive. Instead, (2) is usually written as a chance-constraint $\mathrm{P}(S_1^{\mathbf{t}_1}, S_2^{\mathbf{t}_2}) \leq \delta$, where δ is an upper bound for the collision probability.

When $\mathbf{t}$ is Gaussian distributed, δ has a closed-form solution [13]. The idea is that the integration of a Gaussian distributed variable $\mathbf{t}$ inside a linear half-space can be transformed as a cumulative distribution function (cdf) of a new 1D Gaussian variable [13]:

$$\int_{\mathbf{a}^T\mathbf{t}-b<0} \rho(\mathbf{t}_{21}; \mathbf{p}_{21}, \Sigma_{21}) d\mathbf{t}_{21} = \int_{y<0} \rho(y; p_y, \sigma_y) dy \,,$$

where $\mathbf{t}_{21}$ is the relative position error $\mathbf{t}_{21} = \mathbf{t}_2 - \mathbf{t}_1$, $\mathbf{a}^T\mathbf{t} - b < 0$ defines a linear half-space with the normal $\mathbf{a}$ and the constant b, and $y = \mathbf{a}^T\mathbf{t} - b$ is a 1D variable that follows Gaussian distribution $\mathcal{N}(\mathbf{a}^T\mathbf{p}_{21} - b, \mathbf{a}^T\Sigma_{21}\mathbf{a})$. The cdf result $F_y(0)$ can be easily found by the look-up table.

This means that if the integration region $S_1 \oplus (-S_2)$ in (2) is fully contained by a linear half-space, the collision probability in (2) can be bounded by

$$\mathrm{P}(S_1^{\mathbf{t}_1}, S_2^{\mathbf{t}_2}) = \int_{S_1 \oplus (-S_2)} \rho(\mathbf{t}; \mathbf{p}_{21}, \Sigma_{21}) d\mathbf{t} < F_y(0) \,. \quad (3)$$

The accuracy of the approximation depends on the choice of the linear half-space. Previous methods choose a naive plane with a closed-form solution by using the normalized $\mathbf{p}_{21}$ of the relative position error as the normal $\mathbf{a}$ when objects are spheres or ellipsoids [3] [4]. It is named as *lcc-center* for reference.

Although *lcc-center* is computationally efficient, the geometric models are limited, and the result using such a plane can be conservative. In this work, we propose a linear half-space that gives a more accurate approximation for the collision probability than *lcc-center*.

D. Normal-parameterized superquadric surface

The implicit function of a superquadric is given as:

$$\Psi(\mathbf{x}) = \left(\left(\frac{x_1}{a_1} \right)^{\frac{2}{\epsilon_2}} + \left(\frac{x_2}{a_2} \right)^{\frac{2}{\epsilon_2}} \right)^{\frac{\epsilon_2}{\epsilon_1}} + \left(\frac{x_3}{a_3} \right)^{\frac{2}{\epsilon_1}} = 1$$

A superquadric is a flexible model defined by only five parameters, where the semi-axes $\mathbf{a} = [a_1, a_2, a_3]$ define the size of the superquadric along the principal axes and the epsilons ϵ_1, ϵ_2 control the shape, where $\epsilon_1, \epsilon_2 \in (0, 2)$ ensure the convexity. When $\epsilon_1 = \epsilon_2 = 1$, superquadrics degenerate to be ellipsoids. The closed-form normal-parameterized surface expression $\mathbf{x} = \mathbf{x}(\mathbf{n}) \in \partial S$ can be found in [1] and references therein.

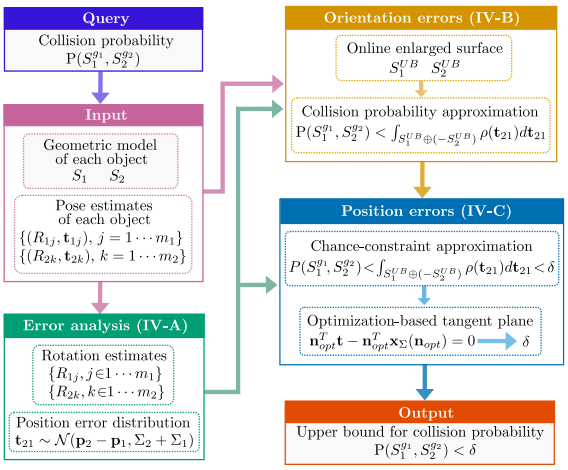


Fig. 2. Proposed probabilistic collision detection method for approximating collision probability between two objects under pose sensing errors.

For two superquadrics S_1 and S_2, given the normal parameterization of $\mathbf{x}_i(\mathbf{n}_i)$, the point on the Minkowski sum boundary $\mathbf{x}_\Sigma \in \partial[S_1 \oplus (-S_2)]$ can be easily found as [1]

$$\mathbf{x}_\Sigma(\mathbf{n}_1) = \mathbf{x}_1(\mathbf{n}_1) - \mathbf{x}_2(-\mathbf{n}_1)$$

because the normals at the contact points of two bodies are anti-parallel, i.e., $\mathbf{n}_1 = -\mathbf{n}_2$, where the normal at $\mathbf{x}_\Sigma$ is the same as the normal at $\mathbf{x}_1$ of S_1.

If applying the rotations $R_1, R_2 \in SO(3)$ to objects, $\mathbf{x}_\Sigma \in \partial[R_1 S_1 \oplus (-R_2 S_2)]$ can be written as $\mathbf{x}_\Sigma(\mathbf{n}) = R_1 \mathbf{x}_1(R_1^T \mathbf{n}) - R_2 \mathbf{x}_2(-R_2^T \mathbf{n})$, where $\mathbf{n}$ is the normal at $\mathbf{x}_\Sigma$, where $R_1^T \mathbf{n}$ and $R_2^T \mathbf{n}$ are normals of points at the original body boundary ∂S_1 and ∂S_2, respectively.

IV. Probabilistic Collision Detection

Let S_1 and S_2 be two rigid bodies. For each body S_i, it has a set of pose estimates $\{g_{ij} = (R_{ij}, \mathbf{t}_{ij}), j = 1 \cdots m_i\}$, with $R_{ij} \in \mathrm{SO}(3)$ and $\mathbf{t}_{ij} \in \mathbb{R}^3$. Due to the imperfect perception module, the pose estimation is inaccurate. The collision probability $\mathrm{P}(S_1^{g_1}, S_2^{g_2})$ under pose estimation errors can be written as:

$$\int_{SE(3)} \int_{SE(3)} \iota(S_1^{g_1} \cap S_2^{g_2} \neq \varnothing) \rho_1(g_1) \rho_2(g_2) dg_1 dg_2 \,, \quad (4)$$

where g_i represents the randomness of the pose of object S_i, distributed according to the density $\rho_i(g_i)$. However, (4) has no closed-form solution. The proposed method for approximating the collision probability is shown in Fig. 2.

A. Error analysis

This work assumes that the position and orientation estimates are independent, i.e., $\rho_i(g_i) = \rho_i(\mathbf{t}_i)\rho_i(R_i)$.

The position error distribution $\rho_i(\mathbf{t}_i)$, i.e., $\mathbf{t}_i \sim \mathcal{N}(\mathbf{p}_i, \Sigma_i)$, can be computed from $\mathbf{t}_{ij}$ of the pose estimates. Because the position estimates between S_1 and S_2 are independent, the

IEEE Robotics and Automation Letters (RA-L) paper, presented at ICRA 2026, Vienna, Austria. Cite as RA-L paper.

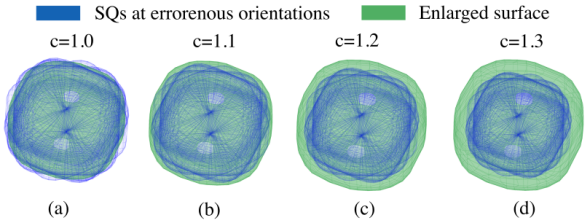


Fig. 3. The enlarged surfaces of S_i^{UB} with different scaling constants c. The boundary lines of the rotated copies are in blue, and the surface and boundary lines of the enlarged surface of S_i^{UB} are in green. This work chooses an empirical value $c = 1.2$ for the real-world experiment.

distribution of the relative position error $\mathbf{t}_{21} = \mathbf{t}_2 - \mathbf{t}_1$ is $\mathbf{t}_{21} \sim \mathcal{N}(\mathbf{p}_{21}, \Sigma_{21})$, where $\mathbf{p}_{21} = \mathbf{p}_2 - \mathbf{p}_1$ and $\Sigma_{21} = \Sigma_2 + \Sigma_1$.

Unlike the position error, the probabilistic model $\rho_i(R_i)$ is not explicitly computed. Given the definition of the mean rotation R_i of the set of rotations $\{R_{ij}, j \in 1 \cdots m\}$ as [14],

$$\sum_{j=1}^{m} \log(R_i^T R_{ij}) = \mathbb{O}\,,$$

where $\log(\cdot) : \mathrm{SO}(3) \to \mathfrak{so}(3)$ is the matrix logarithm. The mean rotation R_i can be solved iteratively.

B. Enlarged surface to handle orientation uncertainties

To handle the orientation uncertainty, we choose to expand the surface of S_i so that the enlarged surface encapsulates its rotated copies R_{ij}. This is inspired by the geometric-based PCD method for position errors [8]. Denoted the enlarged surface of S_i as S_i^{UB}.

To find the enlarged surface of S_i^{UB}, we propose a closed-form normal-parameterized expression of the form:

$$\mathbf{x}_i^{UB}(\mathbf{n}_i) = \frac{c}{m} \left(\sum_{j=1}^{m} R_{ij} \mathbf{x}_i(R_{ij}^T \mathbf{n}_i) \right)\,, \tag{5}$$

where $\mathbf{x}_i^{UB}(\mathbf{n}_i) \in \partial S_i^{UB}$, and $\mathbf{n}_i$ is the normal of ∂S_i^{UB} at $\mathbf{x}_i^{UB}$. m is the number of rotation estimates. c is a scaling constant that can be tuned based on the level of sensing uncertainty in the perception module. (5) can be viewed as the Minkowski sum of m rotated copies of S_i scaled down by m to result in the 'average body' that reflects the contribution of the rotated copies, and scaled up by a constant c to ensure encapsulation. If the object has an exact orientation, the enlarged surface will be the linearly scaled version of itself. In this work, we choose an empirical value of $c = 1.2$. An example of scaling the enlarged surface with different scaling constants is shown in Fig. 3. The enlarged surface is built online for collision checking.

A key property of the encapsulating surface constructed from the rotated copies is that it retains the dominant shape characteristics when the camera is accurate. However, if the perception results are too noisy and the rotation distribution is highly diffuse, the enlarged surface would degenerate to a sphere. But in such a case, no method would perform well.

The Minkowski sum boundary $\mathbf{x}_\Sigma^{UB}(\mathbf{n}) \in \partial[S_1^{UB} \oplus (-S_2^{UB})]$ can be written as:

$$\mathbf{x}_\Sigma^{UB}(\mathbf{n}) = \mathbf{x}_1^{UB}(\mathbf{n}) - \mathbf{x}_2^{UB}(-\mathbf{n})\,. \tag{6}$$

If one body S_i has exact orientation, such as the link of the table-fixed robotic arm, R_{ij} is set to the exact orientation.

With S_1^{UB} and S_2^{UB}, we get an inequality of collision probability (4) as:

$$\mathrm{P}(S_1^{g_1}, S_2^{g_2}) < \int_{\mathbb{R}^3} \iota(\mathbf{t}_{21} \in S_1^{UB} \oplus (-S_2^{UB})) \rho(\mathbf{t}_{21}) d\mathbf{t}_{21}\,. \tag{7}$$

Now, the collision probability under pose sensing uncertainty can be approximated by linearizing the chance constraint problem as shown in (3).

C. Position estimation errors

The quantity in (7) can be further bounded as a chance-constraint problem:

$$\int_{\mathbb{R}^3} \iota(\mathbf{t}_{21} \in S_1^{UB} \oplus (-S_2^{UB})) \rho(\mathbf{t}_{21}) d\mathbf{t}_{21} < \delta\,, \tag{8}$$

where δ is an upper bound and can be solved in closed form. As shown in (3), the accuracy of the upper bound δ for $\mathrm{P}(S_1^{g_1}, S_2^{g_2})$ depends on the choice of the linear half-space.

For a Gaussian-distributed position error $\mathbf{t}_{21}$, the confidence level surface is an ellipsoid, with higher pdf values closer to the center. If the chosen plane is tangent to both the Minkowski sum boundary and the confidence level surface, the resulting half-space contains less of the high pdf region. This provides a better linearization for the chance-constrained PCD problem than the previous *lcc-center*.

To find the tangent half-space, it is important to make the plane tangent to the confidence level surfaces. Because a confidence level surface is an ellipsoid (see Fig. 4 (b)), an easy way to find its tangent plane is to transform the ellipsoid into a sphere. This is done by applying the linear transformation $\Sigma_{21}^{-1/2}$ to the whole space. The transformed position error distribution becomes $\mathbf{t}'_{21} \sim \mathcal{N}(\mathbf{p}'_{21}, \mathbb{I})$, where $\mathbf{p}'_{21} = \Sigma_{21}^{-1/2} \mathbf{p}_{21}$. Given two enlarged superquadrics, their Minkowski sum region $\mathbf{x}_\Sigma^{UB}$ before the linear transformation is shown in the green region in Fig. 4 (b). $\mathbf{x}_\Sigma^{UB}$ after the transformation can be calculated as $\mathbf{x}_\Sigma^{UB'} = \Sigma_{21}^{-1/2} \mathbf{x}_\Sigma^{UB}$, which still has the closed-form solution and remains convex.

The problem of finding the tangent half-space is formulated to find a tangent plane on $\mathbf{x}_\Sigma^{UB'}(\mathbf{n})$ with the shortest distance to the center $\mathbf{p}'_{21}$. This can be formulated as a nonlinear least squares optimization:

$$\min_{\psi} \frac{1}{2} ||\mathbf{p}'_{21} - \mathbf{x}_\Sigma^{UB'}(\mathbf{n}(\boldsymbol{\psi}))||_2^2\,, \tag{9}$$

which is solved by the trust region algorithm.

Because the optimization in (9) is not convex, it does not guarantee the global minimum and is sensitive to the initial value ψ_0. Inspired by [1], this work uses the angular parameter of the center of the position error $\mathbf{p}'_{21}$ as viewed in the body frame of S_1 with the mean rotation R_1, denoted as ${}^1\mathbf{p}'_{21} = R_1^T \mathbf{p}'_{21}$. For 3D cases, ψ_0 equals to:

$$[\mathrm{atan2}\left({}^1\mathbf{p}'_{21,3}, \sqrt{({}^1\mathbf{p}'_{21,1})^2 + ({}^1\mathbf{p}'_{21,2})^2}\right), \mathrm{atan2}\left({}^1\mathbf{p}'_{21,2}, {}^1\mathbf{p}'_{21,1}\right)]$$

After finding the optimized value ψ_{opt}, the normal $\mathbf{a}$ and constant b of the tangent half-space in the untransformed space

IEEE Robotics and Automation Letters (RA-L) paper, presented at ICRA 2026, Vienna, Austria. Cite as RA-L paper.

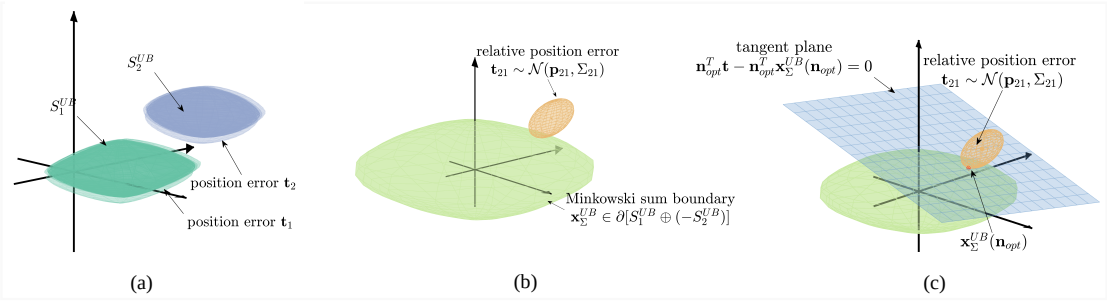


Fig. 4. Illustration of the linear chance constraint method *lcc-tangent*. (a) The two enlarged surfaces S_1^{UB} and S_2^{UB} of two superquadrics S_1 and S_2, and the translated copies of S_1^{UB} and S_2^{UB} with positions sampled based on the distributions of position errors $\mathbf{t}_1$ and $\mathbf{t}_2$; (b) the Minkowski sum boundary $\mathbf{x}_\Sigma^{UB} \in \partial[S_1^{UB} \oplus (-S_2^{UB})]$, and the ellipsoidal level surface of the relative position error $\mathbf{t}_{21} = \mathbf{t}_2 - \mathbf{t}_1$, where $\mathbf{t}_{21} \sim \mathcal{N}(\mathbf{p}_{21}, \Sigma_{21})$; (c) the optimized tangent half-space in the untransformed space based on (9).

are $\mathbf{a} = \mathbf{n}_{\text{opt}}(\psi_{opt})$ and $b = \mathbf{n}_{\text{opt}}^T \mathbf{x}_\Sigma^{UB}(\mathbf{n}_{\text{opt}})$, respectively. Substituting $\mathbf{a}$ and b into (3), the accurate approximation can be calculated, named as *lcc-tangent* for reference.

D. Hierarchical PCD

Although the proposed *lcc-tangent* gives a tight approximation for the collision probability, the existing *lcc-center* for ellipsoids computes faster and can be helpful to screen out low collision probability regions quickly. Therefore, this work proposes a hierarchical PCD method, named *h-lcc*, to balance between computation time and better estimation.

Here the idea of *h-lcc* is explained. For each object, consider a cascade of bounding primitives. The more sophisticated primitive is a tighter fit and the simpler one is looser but less expensive. For example, a superquadric (SQ) is a better fit but more expensive than an ellipsoid (E). If the ellipsoid is an upper bound of the superquadric, which means $SQ_i \subseteq E_i$, then when $P(E_1^{g_1}, E_2^{g_1}) < \delta$, the $P(SQ_1^{g_1}, SQ_2^{g_1}) < \delta$ also holds. In this way, the region where two objects have low collision probability can be quickly screened out if the ellipsoid models do not collide. If not, the more sophisticated model is queried.

V. Results

We first compare the relative goodness of fit using superquadrics and ellipsoids. Then we evaluate the performance, i.e., accuracy and computational time, of the proposed method against existing PCD methods for objects with inaccurate position estimates. In addition, this work assesses the performance of motion planners when using different PCD methods for position errors-only cases. Lastly, a Real2Sim2Real benchmark examines the importance of accounting for pose sensing uncertainty, rather than ignoring the orientation errors of objects in the real-world application. All benchmarks were executed on an Intel Core i9-10920X CPU at 3.5GHz.

A. Evaluation of model fit

To evaluate the relative goodness of fit of using superquadrics and ellipsoids, we compare their performance for approximating 1) convex polyhedra, 2) cuboids, and 3) cylinders. The fitting algorithm for approximating a polyhedron with a superquadric or an ellipsoid is from an existing work [1]. Given the dimension $[l, w, h]$ of a cuboid, the superquadric approximation is with semi-axes $\mathbf{a} = [l/2, w/2, h/2]$ and epsilons $\boldsymbol{\epsilon} = [0.2, 0.2]$. Given the semi-axes $[r_x, r_y]$ and height h of a cylinder, the superquadric approximation is with semi-axes $\mathbf{a} = [r_x, r_y, h/2]$ and epsilons $\boldsymbol{\epsilon} = [0.1, 1.0]$. The ellipsoidal approximation for a cuboid and a cylinder is based on the fitting algorithm, which uses the sampled surface points from the object. The evaluation metric is

$$e = \frac{V(S_{query} \cap S_{true})}{V(S_{query} \cup S_{true})},$$

where $V(\cdot)$ denotes volume, S_{query} is the model with solid body being evaluated (superquadric or ellipsoid), and S_{true} is the exact object (polyhedron, cuboid, or cylinder). A perfect match would give $e = 1$, and a bad match, such as an overly conservative approximation or one with little overlap, will give a small value of e. We generate $N = 100$ random S_{true} and the corresponding candidate model S_{query} and use the average value $\overline{e} = \frac{1}{N}\sum_{i=1}^{N} e_i$ to assess goodness of fit.

Results show that: 1) for convex polyhedra, $\overline{e}_{SQ} = 0.80189$ and $\overline{e}_{Ellip} = 0.66731$; 2) for cuboids, $\overline{e}_{SQ} = 0.95703$ and $\overline{e}_{Ellip} = 0.67732$; 3) for cylinders, $\overline{e}_{SQ} = 0.99396$ and $\overline{e}_{Ellip} = 0.71684$. The statistical results show that a superquadric is a tighter representation than an ellipsoid in general cases for convex bodies, especially for commonly used geometric primitives, i.e., cuboids and cylinders.

B. Benchmark on a single query of PCD

This benchmark compares the proposed PCD methods with state-of-the-art approaches for convex bodies under Gaussian-distributed position errors. Two shape models (ellipsoids and superquadrics) and two error scenarios (one object vs. both objects having independent position errors) yield four test cases: ellipsoids-single-error, superquadrics-single-error, ellipsoids-two-errors, and superquadrics-two-errors.

1) Benchmark setting: Each test generates 100 random object pairs. The semi-axes $[a_1, a_2, a_3]$ of a body are sampled from $(0.2, 1.2)$ m. The epsilon variables are randomly sampled in the range of $(0.01, 0.2)$ for superquadrics. Object centers are sampled in $(0.0, 0.1)$ m and $(0.3, 1.3)$ m, respectively. The orientations are uniformly sampled in $\text{SO}(3)$. Position errors follow a Gaussian distribution with covariance $\Sigma_i = R\Sigma R^T$, where $\Sigma = [4.8, 0, 0; 0, 4.8, 0; 0, 0, 6.0] * 10^{-4}$ is in each object's local frame and $R \in \text{SO}(3)$.

IEEE Robotics and Automation Letters (RA-L) paper, presented at ICRA 2026, Vienna, Austria. Cite as RA-L paper.

TABLE I
ALGORITHMS USED IN THE PCD BENCHMARK.

Notation	Method
Baseline for single error	Monte-Carlo Sampling
Baseline for two errors [9]	Fast Monte-Carlo sampling
EB95 [8]	Bounding volume with 95% of confidence
Divergence [6]	Divergence theorem
lcc-center [4]	Linear chance constraint for ellipsoids
lcc-tangent (ours)	(9)

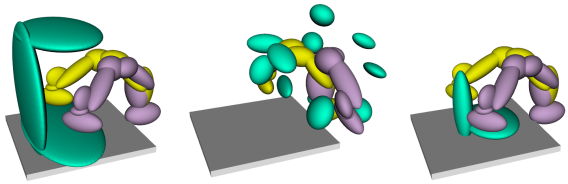
TABLE II
COMPARISON OF PCD METHODS ON ACCURACY AND COMPUTATION TIME UNDER POSITION ESTIMATION ERRORS.

Test setting	PCD method	Mean[3]	Variance[3]	Computation time(s)
ellipsoids single error[1]	EB95	0.0511	0.0310	0.0069
	Divergence	0.0945	0.0651	0.0044
	lcc-center	0.0296	0.0096	**1.7396e-4**
	lcc-tangent (ours)	**0.0162**	**0.0063**	0.0090
superquadrics single error[1]	EB95	0.4885	0.2172	0.0093
	Divergence	0.5072	0.2114	0.0172
	lcc-center	0.5634	**0.1949**	**1.9544e-4**
	lcc-tangent (ours)	**0.4153**	0.2099	0.0143
ellipsoids two errors[2]	EB95	0.1435	0.1014	0.0053
	Divergence	0.1041	0.0713	0.0040
	lcc-center	0.0212	0.0055	**1.4908e-04**
	lcc-tangent (ours)	**0.0142**	**0.0043**	0.0097
superquadrics two errors[2]	EB95	0.2439	0.1597	0.0128
	Divergence	0.1334	0.1231	0.0193
	lcc-center	0.2078	0.1105	**2.1014e-04**
	lcc-tangent (ours)	**0.0226**	**0.0129**	0.0136

[1] Only object S_2 has position estimation error.
[2] Objects S_1 and S_2 have independent position estimation errors.
[3] Mean and variance of the differences (method minus baseline) for each method.

The PCD methods included in the benchmark are summarized in Table I. The baseline for single-error cases employs 10^4 Monte Carlo samples of translated copies of S_2 based on its position error distribution. For double-error cases, a fast Monte Carlo approach generates 10^5 translated copies of object S_2 based on the relative position error distribution [9]. In both cases, the deterministic collision status of translated S_2 with S_1 is computed for each sample. *EB95* is the enlarged bounding volume method with 95% confidence [8]. Divergence [6] applies the divergence theorem to meshed object surfaces (100 points for ellipsoids, 1600 for superquadrics), excluding mesh construction time for fairness of comparison. *lcc-center* uses the linear chance constraint method [3] [4]. As for applying *lcc-center* in superquadrics cases, we first compute the minimum volume enclosing ellipsoids for the superquadrics, and the computation time for computing the bounding ellipsoid is not counted for fairness. *lcc-tangent* is the proposed method of this study in (9).

2) Benchmark results: The results are summarized in Table II. Because all PCD methods are approximations of the collision probability, the results can be higher than 1, while a probability by definition cannot exceed 1. Therefore, if any approximation exceeds 1, we truncate its value to 1. To compare how close the approximation of each PCD method is to the baseline result, we compute the mean and variance of the value of each PCD method result minus the ground truth. A good approximation would give mean and variance close to zero. Overall, *lcc-tangent* is most accurate, closely matching the baseline with lower mean and variance. Compared to *divergence*, which relies on mesh discretization, *lcc-tangent* approximates the collision probability only using one inequality. Although *EB95* can increase accuracy by raising the expansion confidence, this also increases computation time. Meanwhile, *lcc-center* is the fastest since it skips optimization, but it is generally less accurate than *lcc-tangent*, which leverages a plane tangent to both the exact Minkowski sum boundary and the position error's confidence level surface to avoid conservativeness.



(a) (b) (c)

Fig. 5. Three types of environments used in the planning benchmark: (a) clamp, (b) narrow, and (c) sparse. All obstacles are ellipsoids and are subject to position errors. The violet and yellow colors represent the start and goal joint configurations of the arm. Each robot link is bounded by an ellipsoid.

C. Benchmark on planning under position estimation errors

We evaluate how PCD computational accuracy and runtime affect motion planning, comparing *lcc-center* (baseline), *lcc-tangent*, and *h-lcc* used as the sub-modules in RRT-connect (sampling-based) [15] and STOMP (optimization-based) [16] planners.

1) Benchmark setting: The environment settings are shown in Fig. 5. The high-precision industrial robot Franka Emika with the fixed basement is used, and the pose estimation error of the robot can be ignored. The start and goal joint configurations are pre-defined for each setting. Each link of the robot arm is bounded by an ellipsoid, and all obstacles are ellipsoids. In addition, the obstacles are only considered to have position errors, where $\Sigma_i = R\Sigma R^T$, $\Sigma = [4.8, 0, 0; 0, 4.8, 0; 0, 0, 6.0] * 10^{-4}$ in the body-fixed frame, and $R \in \mathrm{SO}(3)$ is the obstacle's orientation. Because all obstacles are static, their position errors will not be propagated. We set the collision threshold to be $\delta = 0.05$. A robot configuration is invalid if any PCD result for a link–obstacle pair exceeds δ. Each environment setup is tested 100 times.

2) Benchmark results: The average path length and planning time for the success cases are listed in Table III.

The narrow environment result shows that with more obstacles in the environment, the planning success rates using our methods will be higher than the baseline. This is because the number of valid robot configurations, for which the collision probability between any of its links and objects is lower than the given threshold ($\delta < 5\%$), is much lower than in a sparse environment. Our methods, especially the *lcc-tangent*, give a tighter approximation of the collision probability than the baseline and thus can help the planner avoid missing valid configurations. By avoiding overly conservative collision checks, our methods tend to find a shorter path in the clamp and the sparse scene.

IEEE Robotics and Automation Letters (RA-L) paper, presented at ICRA 2026, Vienna, Austria. Cite as RA-L paper.

TABLE III
PERFORMANCE OF MOTION PLANNERS WITH DIFFERENT PCD METHODS UNDER POSITION-ONLY UNCERTAINTY.

Motion planner	Environment	PCD method	Planning success rate	Path length (rad)	Planning time (s)
RRTconnect	clamp	lcc-center	100%	5.24	1.28
		lcc-tangent (ours)	100%	4.90	2.62
		h-lcc (ours)	100%	**3.72**	**1.06**
	narrow	lcc-center	8%	**9.08**	**107.78**
		lcc-tangent (ours)	**18%**	9.44	132.92
		h-lcc (ours)	10%	10.52	121.11
STOMP	sparse	lcc-center	100%	8.40	21.04
		lcc-tangent (ours)	100%	**7.15**	17.44
		h-lcc (ours)	100%	**7.15**	**13.06**

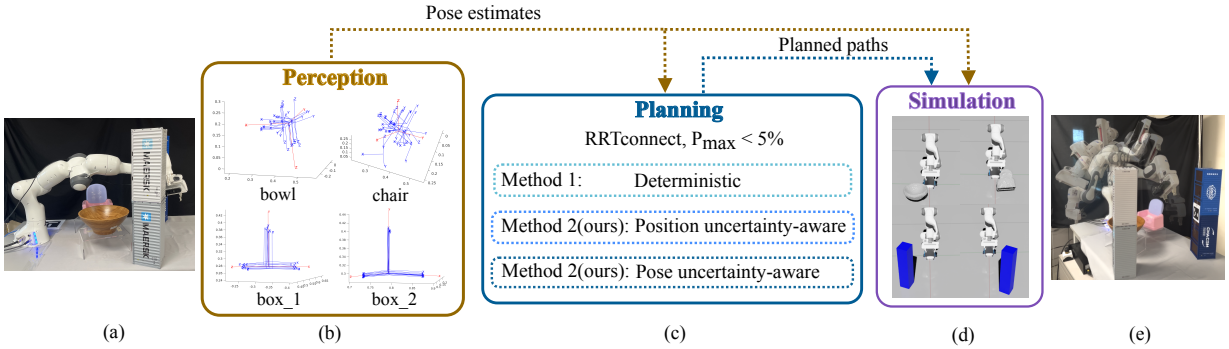


Fig. 6. Real2Sim2Real pipeline to evaluate the reliability of planned paths using different PCD methods. (a) A typical manipulation setting with four objects. (b) Perception results for each object. (c) Motion planning with different levels of uncertainty awareness. (d) Rolling out the possibility of the robot executing the planned path in simulation (Isaac Sim) and calculating the collision probability in execution. (e) A collision-free path in simulation is executed in the real world, which is planned under considering pose sensing uncertainty.

The observed differences in computation time between the two planners stem from their fundamentally different path-planning approaches. RRTconnect relies on random configuration exploration to find a valid path. Since the query time of *lcc-tangent* is longer than that of *lcc-center*, the overall exploration speed of RRTconnect is slower when using *lcc-tangent*. In contrast, STOMP's iterative process allows it to benefit from the precise collision checking provided by *lcc-tangent* and *h-lcc*. The accuracy of these methods enables STOMP to converge faster to an optimized path, leading to reductions in both path length and planning time.

D. Real2Sim2Real evaluation pipeline

This benchmark is designed to test the safety of the planned path in execution when considering different types of sensing uncertainty in motion planning. Here, we create a typical manipulation setting, and the objects' pose estimation is done in the real world. The pose estimation results are used in motion planning when considering no sensing errors (baseline), position estimation errors, and both position and orientation estimation errors. After that, we test the collision probability between the robot and objects in the simulation, where the simulation environment rolls out the possibility of the robot executing the planned path in the real world.

1) Benchmark setting: The experiment setting is shown in Fig. 6 (a). A Franka Emika robot arm moves from a fixed start to a goal configuration. Each robot link is bounded by a superquadric. Object shapes are known, while their poses are sensed via RGB-D cameras: bowl and chair through iterative closest point (ICP), and boxes through ArUco markers. From each pose estimate, we extract the position error distribution and mean orientation.

We run RRTconnect with three collision checkers over 100 trials each: a deterministic checker (*cfc-dist-ls* [1]), a PCD method handling position errors (*lcc-tangent*), and a PCD method for pose errors (encapsulated surfaces + *lcc-tangent*).

Each set of 100 planned paths is executed four times in simulation, once for each of the four objects. At each run, the object's pose is updated randomly from the pose estimations from real-world measurements. Paths are marked "unsafe" if any via point collides with an object, and the collision probability is the fraction of unsafe paths.

2) Benchmark results: The pose estimations for each object are shown as Fig. 6 (b), where the red coordinate is the mean pose, and the blue ones are the pose estimations. ICP yields larger errors for the bowl and chair than ArUco for the boxes. The results are summarized in Table IV. For the position uncertainty-aware PCD method and the pose uncertainty-aware PCD method, if the maximum collision probability approximation between any pair of robotic arm links and objects is larger than the threshold $\delta = 5\%$, then the configuration is invalid. The results show that planners using the three types of PCD methods successfully find a path in each call. In principle, the robot should execute the planned path in the real world with the collision risk being zero or below δ. However, the collision probabilities of path execution in the simulation are 29% with no sensing errors, 9% with only position estimation errors, and 2% with both the position and orientation estimation errors considered in motion planning. The high collision risk is obvious when no sensing

IEEE Robotics and Automation Letters (RA-L) paper, presented at ICRA 2026, Vienna, Austria. Cite as RA-L paper.

TABLE IV
SIMULATED COLLISION RISK OF PATHS GENERATED BY MOTION PLANNERS WITH DIFFERENT UNCERTAINTY AWARENESS.

Collision checker	Planning success rate	Path length (rad)	Planning time (s)	Collision risk in simulation
Deterministic	100%	4.48	0.49	29%
Position uncertainty-aware[1]	100%	4.69	2.77271	9%
Pose uncertainty-aware[1]	100%	4.84	9.62	2%

[1] The maximum collision probability constraint in planning is set to $\delta = 5\%$.

errors are considered. Interestingly, when not considering the orientation errors, the collision probability in execution is larger than the set threshold of 5% in motion planning. Most non-safe cases happen because the robot arm collides with an object when its orientation in simulation deviates greatly from the mean orientation. Thus, the orientation errors are not ignorable even when the perception method is accurate. By creating an encapsulating surface for the object using (5), most surface points of the rotated copies of objects are encapsulated, which is similar to stretching the original body in deviated orientations. The increased planning time when considering the sensing uncertainty is because 1) the single query time of PCD methods is longer than the deterministic method, and 2) more robot configurations in the tree exploration phase are considered invalid.

VI. CONCLUSION

This paper introduces an enhanced PCD method with improved accuracy and robustness for robotic motion planning under sensing uncertainty. By leveraging superquadrics for flexible shape representation and incorporating both position and orientation estimation uncertainties, the proposed approach addresses key limitations in existing PCD methods. A hierarchical PCD method (*i.e., h-lcc*) further ensures computational efficiency by combining fast screening with precise collision probability approximation where needed. Benchmarks demonstrate that the proposed method is twice as close to the Monte-Carlo sampled baseline as the best existing PCD method and reduces path length by 30% and planning time by 37%, respectively. In addition, the Real2Sim2Real pipeline shows that by considering the orientation errors, the collision probability in executing the planned path is only 2%, which is much lower than 9% when considering only the position errors and 29% when ignoring all sensing errors. These advancements enable safer and more efficient motion planning for high-DOF robots in cluttered, unstructured environments.

However, our current framework assumes that the superquadric/ellipsoid models of environmental objects are known a priori and that obstacles are static. In future work, we will integrate online shape approximation to handle previously unseen objects and extend the PCD framework by incorporating trajectory prediction and sensing error propagation to support dynamic obstacles.

REFERENCES

[1] 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*, vol. 7, no. 4, pp. 9485–9492, 2022.

[2] B. Wen, W. Yang, J. Kautz, and S. Birchfield, "Foundationpose: Unified 6d pose estimation and tracking of novel objects," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2024, pp. 17 868–17 879.

[3] H. Zhu and J. Alonso-Mora, "Chance-constrained collision avoidance for mavs in dynamic environments," *IEEE Robotics and Automation Letters*, vol. 4, no. 2, pp. 776–783, 2019.

[4] T. Liu, F. Zhang, F. Gao, and J. Pan, "Tight collision probability for uav motion planning in uncertain environment," in *2023 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2023, pp. 1055–1062.

[5] A. Thomas, F. Mastrogiovanni, and M. Baglietto, "Safe motion planning with environment uncertainty," *Robotics and Autonomous Systems*, vol. 156, p. 104203, 2022.

[6] J. S. Park and D. Manocha, "Efficient probabilistic collision detection for non-gaussian noise distributions," *IEEE Robotics and Automation Letters*, vol. 5, no. 2, pp. 1024–1031, 2020.

[7] N. Vaskevicius and A. Birk, "Revisiting superquadric fitting: A numerically stable formulation," *IEEE transactions on pattern analysis and machine intelligence*, vol. 41, no. 1, pp. 220–233, 2017.

[8] C. Dawson, A. Jasour, A. Hofmann, and B. Williams, "Provably safe trajectory optimization in the presence of uncertain convex obstacles," in *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2020, pp. 6237–6244.

[9] A. Lambert, D. Gruyer, and G. Saint Pierre, "A fast monte carlo algorithm for collision probability estimation," in *2008 10th International Conference on Control, Automation, Robotics and Vision*. IEEE, 2008, pp. 406–411.

[10] N. E. Du Toit and J. W. Burdick, "Probabilistic collision checking with chance constraints," *IEEE Transactions on Robotics*, vol. 27, no. 4, pp. 809–815, 2011.

[11] C. Park, J. S. Park, and D. Manocha, "Fast and bounded probabilistic collision detection for high-dof trajectory planning in dynamic environments," *IEEE Transactions on Automation Science and Engineering*, vol. 15, no. 3, pp. 980–991, 2018.

[12] C. Quintero-Pena, W. Thomason, Z. Kingston, A. Kyrillidis, and L. E. Kavraki, "Stochastic implicit neural signed distance functions for safe motion planning under sensing uncertainty," in *2024 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2024, pp. 2360–2367.

[13] L. Blackmore, M. Ono, and B. C. Williams, "Chance-constrained optimal path planning with obstacles," *IEEE Transactions on Robotics*, vol. 27, no. 6, pp. 1080–1094, 2011.

[14] M. K. Ackerman and G. S. Chirikjian, "A probabilistic solution to the ax= xb problem: Sensor calibration without correspondence," in *International Conference on Geometric Science of Information*. Springer, 2013, pp. 693–701.

[15] I. A. Şucan, M. Moll, and L. E. Kavraki, "The Open Motion Planning Library," *IEEE Robotics & Automation Magazine*, vol. 19, no. 4, pp. 72–82, 12 2012, https://ompl.kavrakilab.org.

[16] M. Kalakrishnan, S. Chitta, E. Theodorou, P. Pastor, and S. Schaal, "Stomp: Stochastic trajectory optimization for motion planning," in *2011 IEEE international conference on robotics and automation*. IEEE, 2011, pp. 4569–4574.