

Distance and Collision Probability Estimation from Gaussian Surface Models

Kshitij Goel and Wennie Tabib

Abstract—This paper describes continuous-space methodologies to estimate the collision probability, Euclidean distance and gradient between an ellipsoidal robot model and an environment surface modeled as a set of Gaussian distributions. Continuous-space collision probability estimation is critical for uncertainty-aware motion planning. Most collision detection and avoidance approaches assume the robot is modeled as a sphere, but ellipsoidal representations provide tighter approximations and enable navigation in cluttered and narrow spaces. State-of-the-art methods derive the Euclidean distance and gradient by processing raw point clouds, which is computationally expensive for large workspaces. Recent advances in Gaussian surface modeling (e.g. mixture models, splatting) enable compressed and high-fidelity surface representations. Few methods exist to estimate continuous-space occupancy from such models. They require Gaussians to model free space and are unable to estimate the collision probability, Euclidean distance and gradient for an ellipsoidal robot. The proposed methods bridge this gap by extending prior work in ellipsoid-to-ellipsoid Euclidean distance and collision probability estimation to Gaussian surface models. A geometric blending approach is also proposed to improve collision probability estimation. The approaches are evaluated with numerical 2D and 3D experiments using real-world point cloud data. Methods for efficient calculation of these quantities are demonstrated to execute within a few microseconds per ellipsoid pair using a single-thread on low-power CPUs of modern embedded computers.

I. INTRODUCTION

Euclidean distance fields and their gradients (often computed using finite differencing) are used for collision avoidance [42] and optimization-based motion planning [47], respectively. These values are calculated by fusing point clouds acquired by range sensors (e.g., LiDARs and depth cameras) and processed on-board the robot. To enable safe navigation outside the sensor field-of-view (FoV), the robot must maintain a map comprised of past fused observations. This is especially important when limited FoV sensors, like depth cameras, are used onboard the robot [54]. Existing Euclidean distance and gradient estimation methods create a spatially discretized map and compute distance and gradient values in each cell via a Breadth-First Search [19], which is computationally expensive in large workspaces or when small grid cells are employed [4]. To overcome the limitations of cell based methods, Gaussian Process (GP) based methods were developed, which implicitly represent the surface and enable continuous-space Euclidean distance and gradient calculation [30, 59, 29]. A GP is regressed from raw point cloud data and used to infer the Euclidean distance and gradient at any test point in

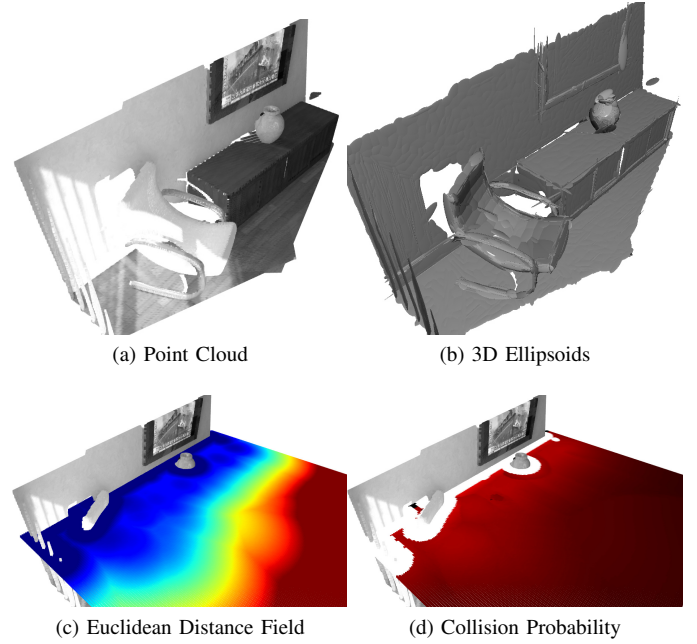


Fig. 1: This work contributes methods to estimate continuous-space collision probability, Euclidean distance and gradient of an ellipsoidal robot body model from a Gaussian surface model (GSM) of the surface. The 3D point cloud shown in (a) is approximated with a GSM, shown as a set of ellipsoids in (b). (c) The Euclidean distance over a 2D slice predicted by the proposed approach is shown as a heatmap (increasing distances from blue to red). (d) The collision probability values (decreasing from red to black, 1.0 in white regions) over the same 2D slice when the robot position is uncertain. *This figure is best viewed in color.*

the workspace. A challenge with these methods is that the training time scales as the size of the input data increases, so optimization-based subsampling must be introduced to enable spatial scalability [58]. In contrast, Gaussian surface models (GSMs) (e.g., mixture models [10, 50, 43, 31] and 3D splatting works [7, 26, 25]) provide relatively compressed and high-fidelity point cloud models. This work leverages the geometric interpretation of GSMs to calculate continuous-space Euclidean distance and gradients.

Collision probabilities are used for motion planning under robot position uncertainty [9, 3]. Like Euclidean distance and gradient field methods, existing methods that estimate collision probability from a surface point cloud utilize probabilistic occupancy queries from a discrete [20, 27] or continuous surface representation [13]. These maps require storing free space cells [11, 21, 1] or raw point clouds [45], leading

to high memory usage. GSMs can alleviate these challenges by effectively compressing point clouds into a finite set of ellipsoids. However, existing occupancy estimation frameworks leveraging GSMs either fit Gaussian components in free space [50, 32] or Monte Carlo ray trace through a local discrete grid [43, 52]. Robot pose uncertainty is not considered. This paper bridges these gaps by proposing a collision probability estimation method, which uses only the surface Gaussians and accounts for Gaussian uncertainty in the robot position.

All the aforementioned techniques require approximating the shape of the robot. Spherical approximations are widely used because inflating the obstacles with the robot size for the purposes of collision avoidance results in a simple configuration space (C-space) [28]. However, recent results in ellipsoidal approximations demonstrate superior planning performance, especially in cluttered and narrow environments [49] for both rigid [34] and articulated robots [23]. These approximations are also used for human bodies in a dynamic environment [61, 35]. Consequently, in this work the proposed methods use an ellipsoidal robot body approximation.

To summarize, given an ellipsoidal robot body model and a set of Gaussians representing the surface point cloud, this paper contributes (Fig. 1) methods to compute the continuous-space

- 1) Euclidean distance estimate between the robot body and surface,
- 2) the approximate gradient of this distance, and
- 3) the upper bound on collision probability when the robot position is a Gaussian random variable.

These methods are evaluated using 2D and 3D surface point clouds from simulation and real-world environments. The software associated with this paper is open-sourced for the benefit of the research community¹.

Section II lists prior work related to the contributions. Section III details the proposed methods. The evaluation is presented in Section IV. Concluding remarks and directions for future work are in Section V.

II. RELATED WORK

Several discrete- and continuous-space methodologies have been proposed to compute Euclidean Distance Fields (EDFs), which indicates distance from an obstacle using only positive values, and Euclidean Signed Distance Fields (ESDFs), which indicate occluded distance using a negative sign. While the applications of these quantities are quite diverse (e.g., shape reconstruction, meshing, etc.), we review continuous-space methods that have been proposed for robot navigation purposes. Collision probability calculation methods under robot position uncertainty are also discussed.

Gaussian Process regression has been used extensively in continuous EDF generation. Lee et al. [30] model the surface point cloud using a GP implicit surface (GPIS) [56], which enables creating continuous ESDFs that are limited to a short-range from the surface. Wu et al. [57] propose the Log-GPIS

map representation that estimates continuous EDFs further into 2D and 3D workspaces. This method models heat diffusion from the surface [6] as a GP regression and uses the logarithm of the regressed heat as the distance. The gradient is also regressed as part of the GP regression. The error in EDF estimates is demonstrated to be lower than the popular discrete-space Voxblox [42] approach. The downside of this method is that the logarithm operation is numerically unstable when the heat approaches zero, which occurs at distances far from the surface. Le Gentil et al. [29] prove that the error in Log-GPIS EDF increases with the distance from the surface and propose an alternate formulation, which leverages reverting functions of covariance kernels for GP regression [46]. While this method achieves EDFs that are relatively accurate, the numerical instability limitation still exists because a logarithm operation is needed over an occupancy value that gets close to zero far from the surface. Furthermore, all of the GP-based methods assume a spherical robot body. In contrast, our approach is numerically stable far from the surface and explicitly accounts for ellipsoidal robot shapes.

GPIS-based representations provide a variance estimate, which indicates the uncertainty in the fused map; however, there remains an open research question about how to compute a collision probability leveraging this variance and the robot position uncertainty. Our approach, however, does provide a collision probability estimate.

Few deep learning approaches enable EDF and gradient estimation. Ortiz et al. [44] model the signed distance function using a 4-layer multi-layer perceptron along with 3D positional embedding for input coordinates [38]. During training, an upper bound on the distance is calculated from sampled free space points by approximating the location of the closest surface points. In larger workspaces, this method requires replaying stored keyframes to avoid catastrophic forgetting. Just like GP-based methods, a spherical robot body is assumed implicitly. Instead of the closest point, we use the notion of the closest ellipsoid on the surface. This ellipsoid belongs to a set of ellipsoids that geometrically approximate the surface GSM. The GSM is created using a methodology that does not suffer from catastrophic forgetting and does not require free space data [15]. Note that the GP-based methods mentioned earlier also do not need to account for catastrophic forgetting and do not use free space points. Due to this reason, the state-of-the-art GP-based method in [29] is used as a baseline method for distance field and gradient comparison (Section IV).

Nguyen et al. [41] utilize deep neural networks to predict the collision probability using a depth image, uncertain partial robot state, and motion primitives as inputs. The network is trained in simulation environments using 1.5 million data points. The Gaussian uncertainty in robot state is propagated through the network through an Unscented Transform [24] of Monte Carlo samples of the Gaussian. The model uncertainty is obtained via the Monte Carlo dropout [14] approach. Both uncertainty calculation methods contain randomness because of the Monte Carlo sampling. Due to this randomness, it is unclear if the uncertainty measure will generalize to real-

¹A link will be added here upon the acceptance of this paper.

world environments that are significantly different from the simulation environments used for training. Furthermore, the training will also require data points obtained at various noise levels in robot state, increasing the cost of prior training. In contrast, the proposed method is geometric and does not require prior training.

There are a few methods to note that enable relevant applications without explicitly creating EDFs. Dhawale et al. [8] demonstrate reactive collision avoidance for a quadcopter from a Gaussian mixture model (GMM) surface model without creating EDFs. The Gaussian components of the GMM are approximated as isocontour ellipsoids. To simplify the C-space [36], a spherical robot approximation is leveraged to inflate the ellipsoids by the radius of the robot body model. Given the inflated ellipsoids, the robot is approximated as a point object and point-in-ellipsoid tests are used for deterministic collision tests. For the same task, Florence et al. [12] provide collision probability estimates for a spherical robot body. Liu et al. [34] use an ellipsoidal model for a quadcopter and use the raw point cloud maps (within a KD-Tree for scalability) for point-in-ellipsoid tests. Srivastava and Michael [50] and Li et al. [32] use surface and free space GMMs to compute continuous-space occupancy prediction. However, the occupancy prediction does not encode robot position uncertainty so that a collision probability may be regressed. In contrast, the proposed methods enable continuous-space collision probability, Euclidean distance and gradient calculation without using free space GMMs for ellipsoidal robots.

III. METHODOLOGY

Starting with the preliminary information Section III-A, the problem statement is provided in Section III-B. The proposed methods are detailed in Section III-C and Section III-D. In this section, small letters are scalars (e.g. x, y), bolded small letters are vectors (e.g. $\mathbf{x}, \mathbf{y}$), capital letters are random variables (e.g. X, Y), capital bolded letters are matrices (e.g. $\mathbf{X}, \mathbf{Y}$), and calligraphic letters are sets (e.g. $\mathcal{X}, \mathcal{Y}$).

A. Preliminaries

A solid ellipsoid of dimension q , center $\mathbf{p} \in \mathbb{R}^q$, and shape $\mathbf{P} \in \mathbb{R}^{q \times q}$ is the quadratic form inequality

$$\mathcal{E}(\mathbf{p}, \mathbf{P}) = \{\mathbf{q} \in \mathbb{R}^q \mid (\mathbf{q} - \mathbf{p})^\top \mathbf{P}(\mathbf{q} - \mathbf{p}) \leq 1\}.$$

The eigen decomposition of $\mathbf{P}$ enables determining the rotation and scale of the ellipsoid. If the eigen decomposition of $\mathbf{P}$ is given by $\mathbf{P} = \mathbf{R}\mathbf{S}\mathbf{R}^\top$, then $\mathbf{R}$ is an orthogonal matrix that provides the ellipsoid's principal axes rotation and $\mathbf{S}$ is a diagonal matrix where entries are the inverse squares of the semi-principal axes lengths. The positive definite matrix $\mathbf{P}^{-1}$ is called the shape matrix of the ellipsoid [18].

The region inside and on a probability isocontour of the probability density function for a q -variate Gaussian random variable X with mean vector $\boldsymbol{\mu}_X$ and covariance matrix $\boldsymbol{\Sigma}_X$ can be geometrically interpreted as an ellipsoid [2, Ch.

2]. The l -th level isocontour is given by the solid ellipsoid $\mathcal{E}_X(\boldsymbol{\mu}_X, \mathbf{R}\boldsymbol{\Gamma}\mathbf{R}^\top)$. The rotation matrix $\mathbf{R}$ contains the eigenvectors of $\boldsymbol{\Sigma}_X^{-1}$ as columns. The entries of the diagonal matrix $\boldsymbol{\Gamma}$ are given by $1/l^2$ times the eigenvalues of $\boldsymbol{\Sigma}_X^{-1}$. For reference, $l = 3$ and 4 provide 99.7% and 99.95% coverage bounds respectively on the points modeled by the random variable X .

If the probability density function is instead given by a weighted sum of Gaussian density functions (i.e., a GMM), then the probability isocontours can be approximated as a set of ellipsoids with one ellipsoid per component of the mixture. Note that this geometric interpretation ignores the weights of the GMM.

B. Problem Statement

A robot $\mathcal{P}$ is equipped with a range sensor (e.g. depth camera, LiDAR, etc.) and it can be moved in a workspace $\mathcal{W} \subseteq \mathbb{R}^q$. In this work q is either 2 or 3. The region inside the workspace occupied by the robot is modeled using the solid ellipsoid $\mathcal{E}_{\mathcal{P}}(\mathbf{p}, \mathbf{P})$. The parameters $\mathbf{p}$ and $\mathbf{P}$ are determined using the Lowner-John ellipsoid fit [48]. The center $\mathbf{p}$ can be uncertain. This uncertainty is modeled using a continuous multivariate Gaussian variable P with the probability density function $\mathcal{N}(\mathbf{p}; \boldsymbol{\mu}_P, \boldsymbol{\Sigma}_P)$. In practice, these uncertainty estimates may be provided by an external state estimation system [39].

The onboard range sensor provides a stream of point cloud measurements $\mathcal{Z}$ of the surfaces $\mathcal{W}_s$ in the workspace. The elements of the point cloud $\mathcal{Z}$ are assumed to be independent and identically distributed samples of a random variable Z with the probability density function given by the GMM

$$f_Z = \sum_{m=1}^M \pi_m \mathcal{N}(\mathbf{z}; \boldsymbol{\mu}_Z^m, \boldsymbol{\Sigma}_Z^m) \quad (1)$$

where π_m , $\boldsymbol{\mu}_Z^m$, and $\boldsymbol{\Sigma}_Z^m$ are the weight, mean, and covariance of the m -th Gaussian component in an M -component GMM. This GMM can be obtained from many recent point cloud modeling methods. The weights may be uniform, such as the GSMs obtained via stochastic gradient-descent [26, 25] or geometric region growing [7]. Alternatively, GMMs with non-uniform weights obtained from Expectation Maximization [15, 50] or scan line segmentation [31] may also be used. The proposed methods in this work only depend on the geometric interpretation of the GMMs and thus do not require the GMMs to be created using a specific method.

Given this information, the following problems are addressed in this work:

- 1) Estimate the closest distance between $\mathcal{E}_{\mathcal{P}}(\mathbf{p}, \mathbf{P})$ and f_Z . Estimate the gradient of this distance at any location. The solution must not depend on a discretization of the space (Section III-C).
- 2) Estimate the collision probability between $\mathcal{E}_{\mathcal{P}}(\mathbf{p}, \mathbf{P})$ and f_Z when the ellipsoid center position is uncertain (Section III-D).

C. Continuous Euclidean Distance Field

The m -th component of the GMM density f_Z from Eq. (1) can be interpreted as an ellipsoid (Section III-A). If it were possible to efficiently calculate the distance of the robot to all M components, we could compute the minimum distance of the robot to the surface GMM. Thus, we require a method to find the closest distance between two ellipsoids. Unfortunately, this distance cannot be derived as a closed-form expression [33].

Many optimization-based approaches have been proposed for this purpose [33, 48, 55]. In this work, the method by Rimón and Boyd [48] is leveraged because it formulates the optimization as an eigenvalue problem. However, the expressions stated in their work require several matrix inversions and square roots. It is not obvious how to avoid these calculations and leverage numerically stable and faster alternatives (e.g. linear system solvers, decompositions) without introducing approximations. The following proposition restates the result from [48] but removes the need to explicitly calculate matrix inversions and square roots.

Proposition 1 (Deterministic Ellipsoid Distance [48]²). Consider two ellipsoids $\mathcal{E}_1(\mathbf{b}, \mathbf{B})$ and $\mathcal{E}_2(\mathbf{c}, \mathbf{C})$. The centers and shapes of these ellipsoids are perfectly known. Let $\mathbf{B} = \hat{\mathbf{B}}\mathbf{\Lambda}_B\hat{\mathbf{B}}^\top$ and $\mathbf{C} = \hat{\mathbf{C}}\mathbf{\Lambda}_C\hat{\mathbf{C}}^\top$ be the eigen decompositions of $\mathbf{B}$ and $\mathbf{C}$ respectively. Consequently, $\mathbf{B}^{1/2} = \hat{\mathbf{B}}\mathbf{\Lambda}_B^{1/2}\hat{\mathbf{B}}^\top$, $\mathbf{B}^{-1} = \hat{\mathbf{B}}\mathbf{\Lambda}_B^{-1}\hat{\mathbf{B}}^\top$, $\mathbf{B}^{-1/2} = \hat{\mathbf{B}}\mathbf{\Lambda}_B^{-1/2}\hat{\mathbf{B}}^\top$, and $\mathbf{C}^{-1} = \hat{\mathbf{C}}\mathbf{\Lambda}_C^{-1}\hat{\mathbf{C}}^\top$. Let λ be the minimal eigenvalue³ of the $2q \times 2q$ matrix

$$\mathbf{M}_1 = \begin{bmatrix} \tilde{\mathbf{C}} & -\mathbf{I}_q \\ -\tilde{\mathbf{c}}\tilde{\mathbf{c}}^\top & \tilde{\mathbf{C}} \end{bmatrix}$$

such that $\mathbf{I}_q$ is an identity matrix of order q , $\tilde{\mathbf{C}} = \mathbf{B}^{1/2}\mathbf{C}^{-1}\mathbf{B}^{1/2}$, and $\tilde{\mathbf{c}}$ is the solution of the linear system

$$(\mathbf{B}^{-1/2}\hat{\mathbf{Q}}\mathbf{\Lambda}_Q^{1/2}\hat{\mathbf{Q}}^\top)\tilde{\mathbf{c}} = \mathbf{c} - \mathbf{b} \quad (2)$$

where $\hat{\mathbf{Q}}\mathbf{\Lambda}_Q\hat{\mathbf{Q}}^\top$ is the eigen decomposition of the real symmetric matrix $\mathbf{B}^{-1/2}\mathbf{C}\mathbf{B}^{-1/2}$. Let μ be the minimal eigenvalue of the $2q \times 2q$ matrix

$$\mathbf{M}_2 = \begin{bmatrix} \mathbf{B}^{-1} & -\mathbf{I}_q \\ -\tilde{\mathbf{b}}\tilde{\mathbf{b}}^\top & \mathbf{B}^{-1} \end{bmatrix}$$

such that $\tilde{\mathbf{b}} = -\lambda\mathbf{B}^{-1/2}\alpha$ and α is the solution to the linear system

$$\{\mathbf{B}^{-1/2}(\lambda\mathbf{I}_q - \tilde{\mathbf{C}})\mathbf{B}^{1/2}\}\alpha = \mathbf{c} - \mathbf{b}. \quad (3)$$

Given these quantities, the closest distance estimate $d(\mathcal{E}_1, \mathcal{E}_2)$ between $\mathcal{E}_1$ and $\mathcal{E}_2$ is

$$d(\mathcal{E}_1, \mathcal{E}_2) = \|\mathbf{d}^*\|, \quad (4)$$

²The typographical errors in the expressions for $\tilde{\mathbf{b}}$ and $\mathbf{y}^*$ in [48, Prop. 3.2] have been corrected here.

³The minimal eigenvalue is equal to the real part of the eigenvalue with the lowest real part amongst all eigenvalues (real or complex). This notion is required because the eigenvalues of $\mathbf{M}_1$ and $\mathbf{M}_2$ may be complex numbers.

where $\|\cdot\|$ denotes the L2-norm of a vector and $\mathbf{d}^*$ is the solution to the linear system

$$(\mu\mathbf{I}_q - \mathbf{B}^{-1})\mathbf{d}^* = -\mu\lambda\alpha. \quad (5)$$

Note that calculating matrix inverses and square roots of diagonal matrices like $\mathbf{\Lambda}_B$ only requires inverse and square root operations on their scalar diagonal entries as opposed to full matrix operations. Therefore, there are no explicit matrix inversion or matrix square root calculations required in Proposition 1.

Using Proposition 1, the distance between the robot body ellipsoid $\mathcal{E}_P(\mathbf{p}, \mathbf{P})$ and the surface model f_Z can be computed by computing the minimum over all M components of f_Z :

$$d^*(\mathcal{E}_P(\mathbf{p}, \mathbf{P}), f_Z) = \min_m d(\mathcal{E}_P(\mathbf{p}, \mathbf{P}), \mathcal{E}_m) \quad (6)$$

where $d(\cdot)$ is the distance function from Eq. (4) and $\mathcal{E}_m$ denotes the ellipsoid corresponding to the m -th Gaussian component in the GMM density f_Z . This ellipsoid can be constructed for isocontours of the Gaussian component Section III-A.

To calculate the gradient, Rimón and Boyd [48] suggest deriving the analytical gradient of Eq. (4) as it is differentiable. However, computing this gradient requires several matrix multiplications and inversions. To save computational resources onboard robots, an approximation is leveraged. From Eq. (6), we also get the ellipsoid on the surface, $\mathcal{E}_m^*$, that is closest to the robot. The gradient vector is approximated using the position vector $\mathbf{d}^*$ (from Eq. (4)) for $\mathcal{E}_m^*$ and $\mathcal{E}_P$,

$$\nabla d^*(\mathcal{E}_P(\mathbf{p}, \mathbf{P}), f_Z) = \frac{\mathbf{d}^*}{\sqrt{\mathbf{d}^{*\top}\mathbf{d}^*}}. \quad (7)$$

Under the limiting condition where the number of components M is equal to the number of points in the point cloud $\mathcal{Z}$ such that each point in the point cloud is represented with a Gaussian component, the distance (Eq. (6)) and gradient (Eq. (7)) formulations are exact (i.e., ground truth values). As M decreases, the error in d^* and ∇d^* increases relative to the ground truth value.

The computation in Eq. (6) scales linearly with the number of components M in the GMM. Local submap extraction approaches such as hash maps [15] or spatial partitioning data structures such as KD-tree [22] and B+-tree [40] may enable improved scalability as M increases.

D. Collision Probability Under Position Uncertainty

The following corollary to Proposition 1 will be used in this section.

Corollary 1 (Deterministic Ellipsoid Collision Check [53]). If it is desired to only check whether $\mathcal{E}_1$ and $\mathcal{E}_2$ collide (under deterministic conditions), an explicit calculation of Eq. (4) is not required. When $\mathcal{E}_1$ and $\mathcal{E}_2$ touch or intersect, Thomas et al. [53] show that $\mathbf{y}^\top \mathbf{D}^\top \mathbf{B} \mathbf{D} \mathbf{y} \leq 1/\lambda^2$, where $\mathbf{y} = \mathbf{c} - \mathbf{b}$, $\mathbf{D} = \mathbf{B}^{-1/2}(\lambda\mathbf{I}_q - \tilde{\mathbf{C}})^{-1}\mathbf{B}^{1/2}$, and other quantities are as calculated in Proposition 1. We simplify this inequality to

$$\mathbf{y}^\top \mathbf{B}^{1/2} \mathbf{A}^{-1} \mathbf{B}^{1/2} \mathbf{y} \leq 1/\lambda^2 \quad (8)$$

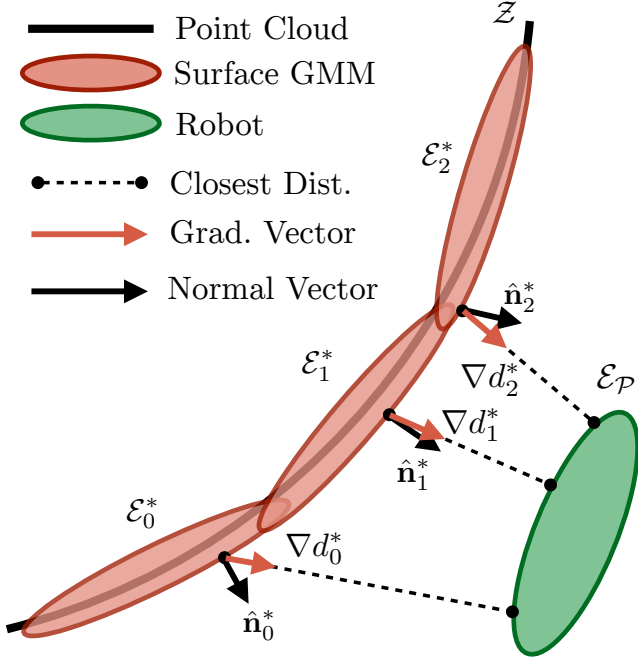


Fig. 2: Illustration of the quantities required for blending weights calculation in collision probability estimation (Section III-D). For each of the ellipsoids on the surface, the blending weight w_k is the dot product of the distance gradient ∇d_k^* and the normal $\hat{\mathbf{n}}_k^*$. This figure is best viewed in color.

where $\mathbf{A} = (\lambda \mathbf{I}_q - \tilde{\mathbf{C}})^2$ is a real symmetric matrix. It is shown in [48, Thm. 2] that when the center of ellipsoid $\mathcal{E}_1$, $\mathbf{b}$, lies outside of $\mathcal{E}_2^*$, λ is always negative. Since $\tilde{\mathbf{C}} = \mathbf{B}^{1/2} \mathbf{C}^{-1} \mathbf{B}^{1/2}$ with $\mathbf{B} \succ 0$ and $\mathbf{C} \succ 0$, it follows that $\tilde{\mathbf{C}} \succ 0$. Therefore, the matrix $\lambda \mathbf{I}_q - \tilde{\mathbf{C}} \prec 0$, which implies $\mathbf{A} \succ 0$ (i.e., $\mathbf{A}$ is positive definite). Therefore, the Cholesky decomposition of $\mathbf{A}$ can be used to efficiently compute Eq. (8) without explicit matrix inversions or square roots.

Let $\mathbf{v} = \mathbf{y}^\top \bar{\mathbf{A}} \mathbf{y}$ where $\bar{\mathbf{A}} = \mathbf{B}^{1/2} \mathbf{A}^{-1} \mathbf{B}^{1/2}$. When the center $\mathbf{b}$ of the ellipsoid $\mathcal{E}_1$ is a Gaussian distributed random variable with density $\mathcal{N}(\mathbf{b}; \boldsymbol{\mu}_B, \boldsymbol{\Sigma}_B)$, $\mathbf{v}$ becomes a random variable as both $\mathbf{y}$ and $\bar{\mathbf{A}}$ depend on $\mathbf{b}$. Under a conservative assumption that $\bar{\mathbf{A}}$ and λ are deterministic (calculated using the mean $\boldsymbol{\mu}_B$)⁵, it is proved in [53] that $P_B(\mathcal{E}_1, \mathcal{E}_2) \equiv P_B(\mathbf{v} \leq 1/\lambda^2)$ and an upper bound on the probability can be estimated via

$$P_B(\mathbf{v} \leq 1/\lambda^2) \leq \frac{\eta \sqrt{\mathbb{V}[\mathbf{v}]}}{\mathbb{E}[\mathbf{v}] + \eta \sqrt{\mathbb{V}[\mathbf{v}]} - (1/\lambda^2)} \quad (9)$$

where $\mathbb{E}[\mathbf{v}]$ and $\mathbb{V}[\mathbf{v}]$ denote the expectation and variance of $\mathbf{v}$, λ is as defined in Proposition 1, and η is a constant. The values $\mathbb{E}[\mathbf{v}]$ and $\mathbb{V}[\mathbf{v}]$ can be exactly calculated using [37,

⁴This assumption does not lead to a loss of generality because when $\mathbf{b}$ lies inside $\mathcal{E}_2$, the ellipsoids are definitely intersecting.

⁵Liu et al. [35] propose leveraging the Minkowski sum of ellipsoids to relax this assumption. However, it is computationally difficult to obtain a tight approximation of this sum [18].

Thm. 3.2b.2]

$$\begin{aligned} \mathbb{E}[\mathbf{v}] &= \text{tr}[\bar{\mathbf{A}} \boldsymbol{\Sigma}_B] + (\mathbf{c} - \boldsymbol{\mu}_B)^\top \bar{\mathbf{A}} (\mathbf{c} - \boldsymbol{\mu}_B) \text{ and} \\ \mathbb{V}[\mathbf{v}] &= 2 \text{tr}[(\bar{\mathbf{A}} \boldsymbol{\Sigma}_B)^2] + 4(\mathbf{c} - \boldsymbol{\mu}_B)^\top \bar{\mathbf{A}} \boldsymbol{\Sigma}_B \bar{\mathbf{A}} (\mathbf{c} - \boldsymbol{\mu}_B) \end{aligned}$$

where $\text{tr}[\cdot]$ denotes the trace of a matrix. Note that explicit matrix square roots are not required in this upper bound calculation (Proposition 1). Estimation of the constant η is difficult. Thomas et al. [53] set this value to 0.25 for all cases⁶. However, we found that in some cases this value leads the denominator in Eq. (9) to be negative (i.e., $\mathbb{E}[\mathbf{v}] + \eta \sqrt{\mathbb{V}[\mathbf{v}]} < (1/\lambda^2)$). Therefore, in practice when this denominator is negative, starting with $\eta = 0.25$, we keep increasing it by 0.5 until the denominator turns positive. This is a valid approximation because η is used to upper bound the value of $\mathbf{v}$ [53]; increasing η makes this bound loose for certain ellipsoid pairs but maintains the validity of Eq. (9).

To compute the probability in Eq. (9) relative to the surface GMM f_Z , one approach is to identify the surface ellipsoid closest to the robot ($\mathcal{E}_m^*$) from Eq. (6) and evaluate Eq. (9) between the robot ellipsoid $\mathcal{E}_P(\mathbf{p}, \mathbf{P})$ and $\mathcal{E}_m^*$. However, this approach yields a non-smooth collision probability field due to a rapid spatial change in $\mathcal{E}_m^*$. Such a probability field may not be useful for optimization-based motion planning under uncertainty [61].

To mitigate this problem, we propose blending the collision probabilities of the K nearest surface ellipsoids from the robot ellipsoid. Such neighbors can be efficiently queried from spatial partitioning data structures such as KD-Tree [22]. In this work, we empirically set $K = 3$ for 2D and $K = 9$ for 3D workspaces. The final probability is given by

$$P^*(\mathcal{E}_P(\mathbf{p}, \mathbf{P}), f_Z) = \frac{1}{\sum_k w_k} \sum_k w_k P(\mathcal{E}_P(\mathbf{p}, \mathbf{P}), \mathcal{E}_k^*), \quad (10)$$

where w_k denotes the blending weight of the k -th ellipsoid $\mathcal{E}_k^*$ in the set of nearest K ellipsoids $\{\mathcal{E}_1^*, \dots, \mathcal{E}_K^*\}$. The blending weight w_k is the dot product of ∇d_k^* and the eigenvector directed towards the robot position, $\hat{\mathbf{n}}_k^*$, corresponding to the minimum eigenvalue of the shape matrix (Fig. 2). The dot product will be negative in cases when the angle between $\hat{\mathbf{n}}_k^*$ and ∇d_k^* is in the range $[\pi/2, 3\pi/2]$. For example, this situation can arise when ellipsoids represent a sharp turn. In such cases, we ignore the contribution of the ellipsoid to the collision probability (i.e., $w_k = 0$).

IV. RESULTS

This section presents an evaluation of the computational cost and accuracy of the proposed methods.

Targeting single-threaded operation on a CPU, the proposed methods are implemented in C++ using the Eigen library [17] with the highest level of optimization enabled in the GNU GCC compiler (`-march=native -O3`). Python bindings are created using `pybind11` for easy integration, testing, and visualization. The performance of this implementation is

⁶There is a typographical error in [53] which states that $\eta = 1$. The tight upper bound calculated in Section IV-D of [53] is correct when $\eta = 0.25$.

measured on several desktop and embedded platform CPUs to reflect the applicability of the proposed methods for a wide range of robots (Section IV-A).

The accuracy of the proposed continuous distance field is compared with the state-of-the-art continuous GP-based approach by Le Gentil et al. [29]. The accuracy measures for both the distance and gradient prediction are detailed in Section IV-B. For the proposed collision probability calculation method, the improvement due to the blending approach (Eq. (10)) is analyzed for different levels of noise in robot position. Experiments are conducted using 2D (Section IV-C) and 3D (Section IV-D) point clouds. For 3D, both simulated and real-world point clouds are used for evaluation.

A. Computational Performance

For this analysis, 100000 pairs of random ellipsoids are created by randomizing axis lengths, positions, and rotations of 3D ellipsoids. Axis lengths and positions (in all directions) are sampled uniformly from the intervals $[0.1, 0.5]$ m and $[-10, 10]$ m, respectively. Random rotation matrices are generated using the Haar distribution [51]. Time taken for ellipsoid pair data structure initialization, distance and gradient computation, and collision probability calculation is measured. The initialization procedure caches matrices $\hat{\mathbf{B}}, \mathbf{A}_B, \mathbf{B}^{1/2}, \mathbf{B}^{-1/2}$, and $\mathbf{B}^{-1}$ for $\mathcal{E}_1(\mathbf{b}, \mathbf{B})$ (and the respective counterparts for $\mathcal{E}_2(\mathbf{c}, \mathbf{C})$), as defined in Proposition 1. Distance and gradient computation times correspond to calculating Eq. (4). For collision probability calculation, Eq. (9) is used. The performance is measured on one desktop (Intel i9-10900K) and three embedded platforms (NVIDIA Orin AGX, Orin NX, and Orin Nano).

Mean and standard deviation statistics of elapsed time (in microseconds) are reported in Table I. It can be concluded that the time taken by all computers in Table I is between 10 to 60 microseconds. The total time (last column) represents the worst case where new ellipsoids are initialized every time the distance, gradient, and collision probability are calculated. In practice, the ellipsoids are initialized and updated using a few sensor observations; not for every distance or collision probability computation. As expected, the computation time increases as the clock speed of the CPU decreases.

Since real-world motion planning systems largely run on CPUs, these observations indicate that the proposed methods may be beneficial for adapting many existing motion planners towards leveraging GSMs as the environment representation. Furthermore, since the proposed methods run efficiently on a CPU, the GPU can be used for tasks like online high-fidelity RGB-D reconstruction. Note that fast fitting of GSMs to point cloud data is an active area of research, with recent approaches demonstrating high frame-rate operation on embedded computers like the NVIDIA TX2 [32].

B. Accuracy Measures

The accuracy in the distance field is measured (in meters) using the Root-Mean-Squared-Error (RMSE) between the prediction and the ground truth. A lower RMSE indicates

Device (CPU)	Time Taken (10^{-6} s), single-threaded execution			
	Init.	Dist. + Grad.	Coll. Prob.	Total
i9 3.7 GHz	9.0 ± 1.7	9.5 ± 2.0	2.2 ± 0.9	20.7 ± 3.0
AGX 2.2 GHz	18.4 ± 1.1	17.3 ± 1.8	5.3 ± 0.6	41.0 ± 2.3
NX 2.0 GHz	22.4 ± 4.2	22.9 ± 2.6	6.7 ± 0.8	52.0 ± 5.2
Nano 1.5 GHz	24.5 ± 1.6	24.8 ± 2.7	7.0 ± 0.8	56.2 ± 3.4

TABLE I: Mean and standard deviation of the time taken (in microseconds) to initialize a pair of ellipsoids and estimate distance, gradient, and collision probability using single-threaded execution on various CPUs.

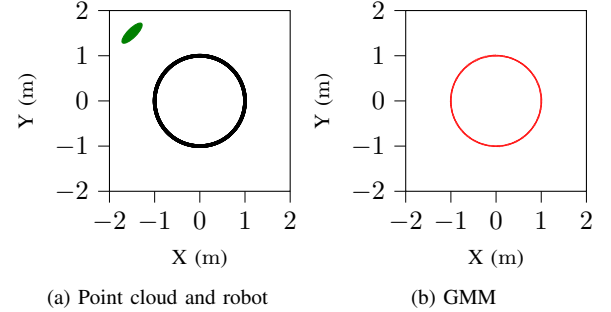


Fig. 3: Two-dimensional numerical scenario used in Section IV-C. (a) The input point cloud along with the representative position of the robot body model (green ellipse). (b) The GMM of the point cloud.

higher accuracy. For the gradient of the distance field, we use $1.0 - \text{RMSE}(\cos(\mathbf{D}_p, \mathbf{D}_g))$, where the cosine is computed at each corresponding point in the predicted ($\mathbf{D}_p$) and the ground truth vector field ($\mathbf{D}_g$) [44, Eq. 15]. This score is referred to as the Cosine Error Score (CES) in the following sections. A lower CES implies a better alignment between the predicted and the ground truth gradient.

The ground truth distance between ellipsoid and point cloud is computed using the point cloud distance between points densely sampled on the robot ellipsoid surface and the surface point cloud. For 2D numerical experiments, the ground truth gradient is calculated using the finite difference method on the ground truth distance field. For 3D experiments, the ground truth gradient vector is given by the surface normal at the point in the surface point cloud that is closest to the robot point cloud.

l	Baseline [29]		σ	M	Proposed	
	RMSE (m) ↓	CES ↓			RMSE (m) ↓	CES ↓
0.5	0.073	0.003	0.1	40	0.007	0.003
0.4	0.080	0.003	0.2	19	0.030	0.009
0.3	0.093	0.003	0.3	14	0.055	0.017
0.2	0.103	0.003	0.4	12	0.072	0.018
0.1	0.111	0.003	0.5	7	0.186	0.044

TABLE II: Quantitative analysis for errors in EDF and gradient at different hyperparameter values. The best RMSE and CES values are bolded. The proposed method enables higher EDF accuracy compared to the baseline.

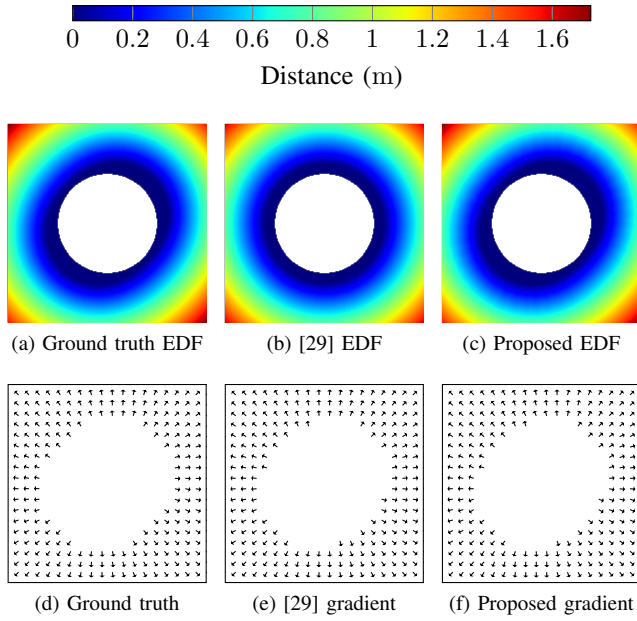


Fig. 4: Comparison of EDF and gradient accuracy for the 2D scenario in Fig. 3a. Qualitative difference for EDF can be observed in (a), (b), and (c) and for the gradient in (d), (e), and (f). *This figure is best viewed in color.*

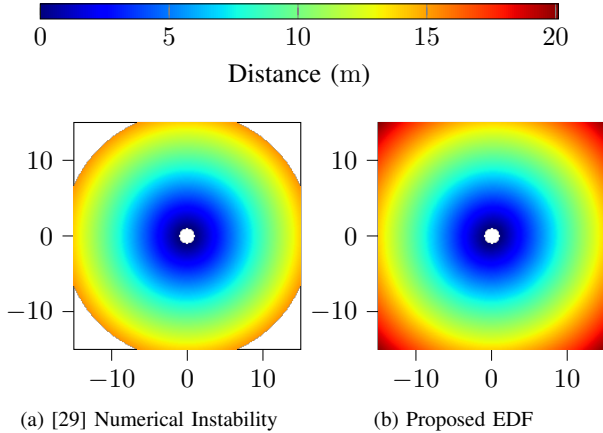


Fig. 5: Heatmaps for EDF far from the surface generated using the baseline (a) and proposed (b) methods. The blank space at the edges in (a) shows the region where the EDF is undefined. Contrary to the baseline, the proposed approach is numerically stable in large workspaces.

C. 2D Experiments

The experimental setup is shown in Fig. 3. A point cloud with 1000 points is generated by uniformly sampling the boundary of a circular obstacle of radius 1.0 m. This point cloud represents a set of range measurements of the obstacle. The robot body is given by an ellipsoid with semi-axes lengths (0.3, 0.1) meters and a rotation by 45 degrees from the positive x -axis (Fig. 3a). The GMM shown in Fig. 3b contains $M = 40$ components and it is generated from the point cloud using the approach from [16].

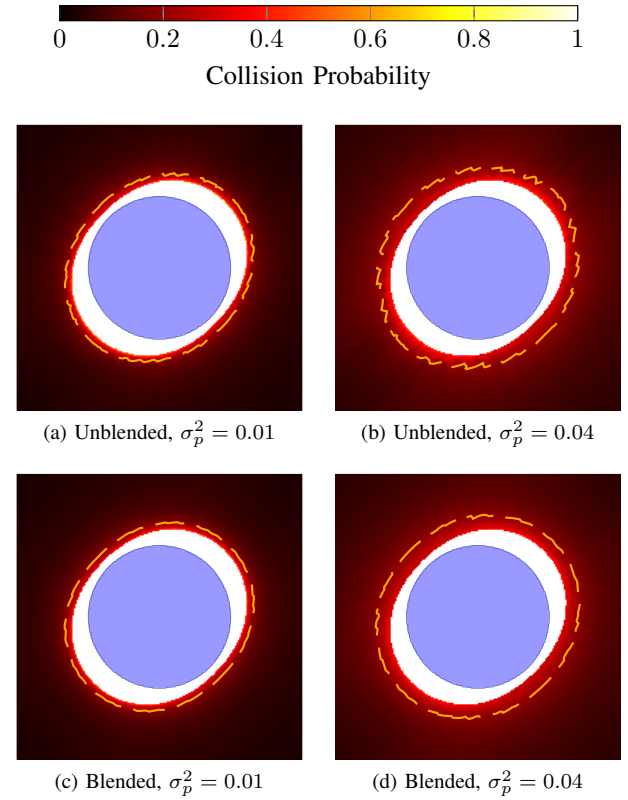


Fig. 6: Heatmaps for collision probabilities computed using the proposed approach at different noise levels. The blue circle is the obstacle surface from Fig. 3a. From (a) to (b) and from (c) to (d) the noise increases and so do the collision probabilities (larger red regions in (b) and (d)). The dashed orange lines are isocontours at 15% collision probability. In the blended case the isocontour lines are smoother compared to the unblended approach. *This figure is best viewed in color.*

The number of components M may vary with the bandwidth hyperparameter σ in [16]; therefore, in quantitative results five different bandwidth values are considered. The baseline approach [29] uses the raw point cloud in all cases but requires a hyperparameter called characteristic length l . It is stated in [29] that specifying this parameter is an open area of research. Therefore, the RMSE and CES values are computed for a set of σ and l parameters.

The center position of the robot body ellipsoid is varied on a 200×200 uniform grid in a workspace with extents $[-2.0, 2.0]$ meters in both x and y directions. At each robot position the Euclidean distance and its gradient from the surface point cloud are measured, resulting in EDFs and gradient vector fields in the 2D workspace for the ground truth, baseline, and proposed methods (Fig. 4).

Qualitatively, the distance isocontours appear rotated by 45 degrees in the ground truth EDF (Fig. 4a) because of the ellipsoid's fixed orientation. The baseline approach implicitly assumes a circular robot body of radius equal to the semi-major axis length of the ellipsoid (0.3 m). This results in a conservative EDF estimate (Fig. 4b) relative to the ground

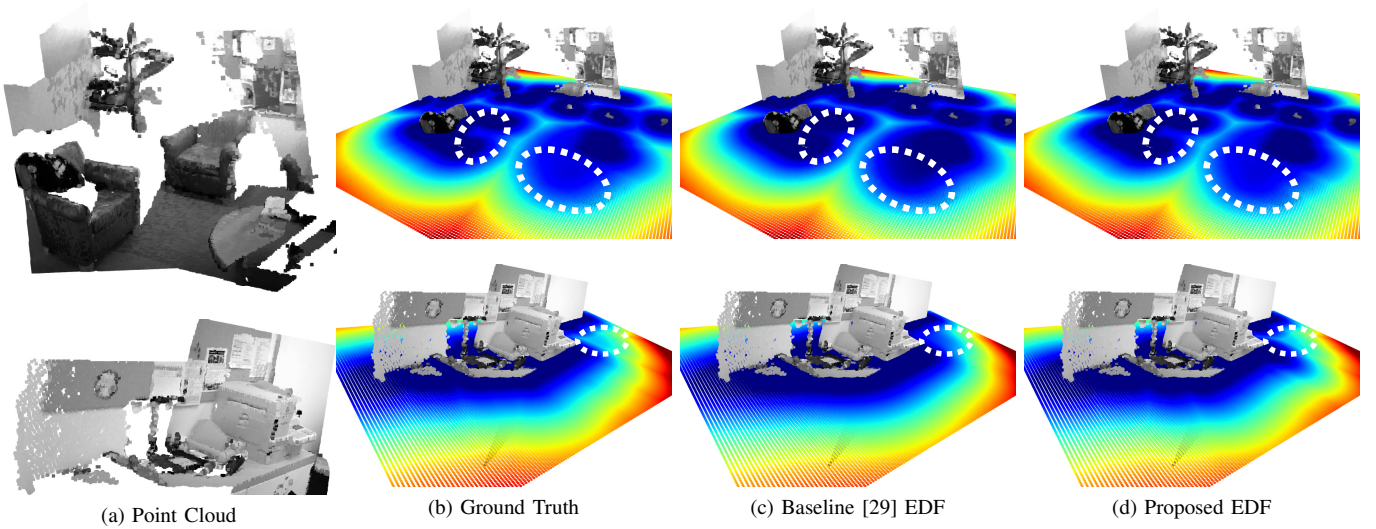


Fig. 7: Heatmaps for (b) ground truth, (c) baseline [29], and (d) proposed EDFs generated using the real-world 3D point cloud shown in (a). Note the difference in baseline and proposed EDFs relative to the ground truth in the dashed white regions. The dark blue regions are bigger in the baseline demonstrating conservative EDF estimation due to an implicit spherical robot body assumption. The proposed approach accounts for the ellipsoidal robot body while enabling continuous-space queries. *This figure is best viewed in color.*

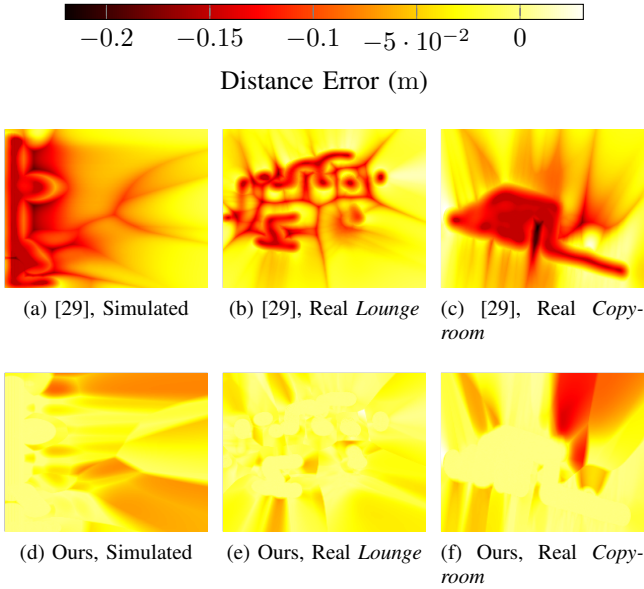


Fig. 8: Distance field estimation error heatmaps for the baseline and proposed approaches. The proposed approach enables a lower estimation error compared to the baseline.

truth. However, the proposed approach (using the GMM in Fig. 3b) accounts for the ellipsoid robot model explicitly, resulting in a relatively accurate EDF (Fig. 4c). Notably, this EDF accuracy improvement is achieved while maintaining a similar level of accuracy in the gradient of EDF (see the arrows in Figs. 4d to 4f).

The quantitative results are summarized in Table II. In the baseline case, for decreasing l the EDF accuracy increases but the gradient accuracy is observed to be constant. For the proposed method, the EDF and gradient accuracy increases

with decreasing σ . The best performing case for the proposed method enables about $10\times$ more accurate EDF prediction than the best performing baseline while achieving the same gradient accuracy. If a higher σ is used, the EDF estimates remain conservative so the robot still remains safe.

Figure 5 provides a comparison of the numerical stability during EDF computation far from the surface. The baseline approach requires a logarithm computation of an occupancy value that gets close to zero as the distance from the surface increases. Therefore, at a certain distance the estimates are invalid as $\log(0) \rightarrow -\infty$. In Fig. 5a, this effect can be seen in the blank areas nearly 10m away from the surface where the EDF is not defined. In contrast, the proposed approach can estimate distance everywhere in the workspace as it depends on the distance between ellipsoids (Fig. 5b).

Let the center of the robot ellipsoid be Gaussian-distributed with the spherical covariance $\sigma_p^2 \mathbf{I}_2$, where $\mathbf{I}_n$ denotes the identity matrix of order n . The unblended and blended collision probabilities for $\sigma_p^2 = 0.01$ and 0.04 along with isocontours for probability level 0.15 are shown in Fig. 6. It is observed that for increasing levels of noise there is an overall increase in collision probabilities and the blending approach yields a smoother collision probability field. Both observations imply that the blended collision probability calculation method may be used for continuous-space queries in uncertainty-aware motion planning frameworks [61, 35].

D. 3D Experiments

Three point clouds are used in 3D experiments: a simulated point cloud from the Living Room dataset [5], a real-world point cloud from the Lounge dataset [60], and a real-world point cloud from the Copyroom dataset [60]. The point clouds are constructed using two 320×240 RGB-D frames from each dataset (see Figs. 1a and 7a). The accuracy of EDF and its

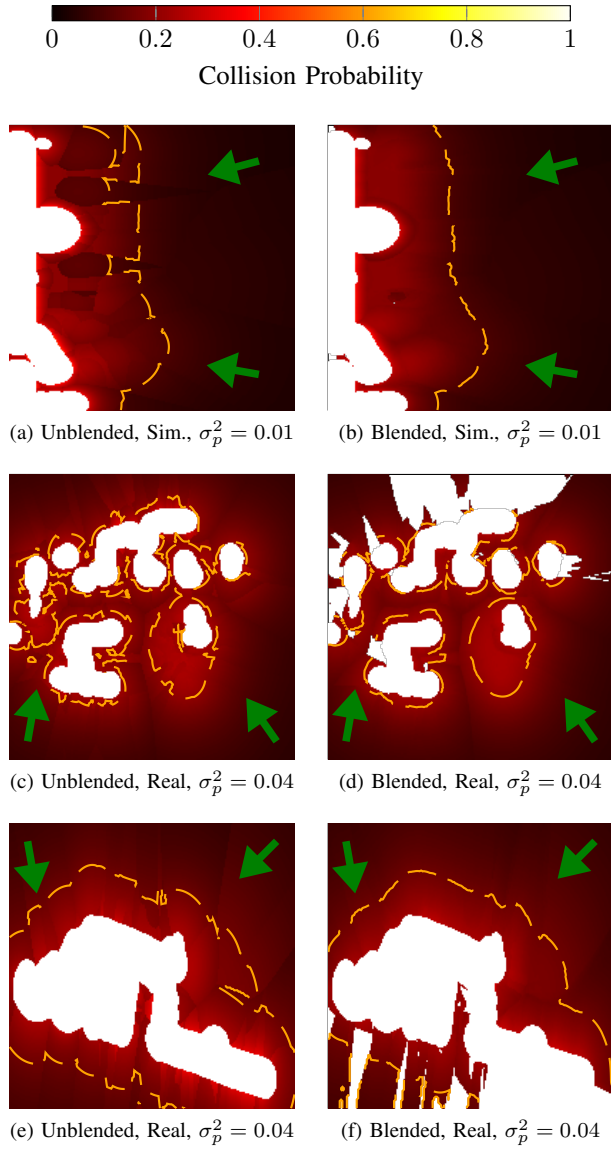


Fig. 9: Unblended and blended collision probability fields over 2D slices of 3D simulated and real-world point clouds. More noise in robot position is added to the real-world cases to simulated the effect of higher position uncertainty during real-world deployments. Dashed lines show 10% probability isocontours. Green arrows show the directions of camera frustums from which the surface point cloud data is collected. The blending approach produces smoother collision probability estimates while ignoring occluded regions. *This figure is best viewed in color.*

gradient is studied for 2D uniform grid (200×200) slices of these environments, as done in prior work [59, 29]. The robot is a 3D ellipsoid with semi-axis lengths (0.15, 0.15, 0.07) and it is rotated about the z -axis by 45 degrees. Figures 1c and 1d show EDF and collision probability outputs for the proposed approach on the simulated point cloud.

Figure 7 shows a qualitative comparison of the EDF obtained from different methods for the real point clouds. We observe the same outcome as in the 2D experiments; the pro-

	Simulated		Lounge		Copyroom	
Approach	RMSE ↓	CES ↓	RMSE ↓	CES ↓	RMSE ↓	CES ↓
[29] (0.1)	0.054	0.166	0.039	0.258	0.056	0.277
[29] (0.2)	0.084	0.168	0.061	0.258	0.075	0.278
[29] (0.3)	0.119	0.173	0.098	0.258	0.100	0.283
[29] (0.4)	0.145	0.171	0.132	0.257	0.129	0.290
Ours (0.02)	0.042	0.159	0.023	0.263	0.046	0.289
Ours (0.03)	0.038	0.164	0.042	0.263	0.058	0.295
Ours (0.04)	0.068	0.181	0.038	0.267	0.057	0.305
Ours (0.05)	0.085	0.191	0.054	0.267	0.062	0.296

TABLE III: Quantitative results using simulated and real 3D point clouds at different hyperparameter settings for the baseline and proposed methods. The best RMSE and CES values for each dataset are bolded.

posed approach is relatively accurate because it accounts for the ellipsoidal robot body explicitly. For a better visualization of the difference in errors incurred by the two methods, Fig. 8 contains error heatmaps for the EDFs in Fig. 7.

For different values of hyperparameters the quantitative comparison of EDFs and gradient is summarized in Table III. The range of hyperparameter l for the baseline approach is chosen by grid search to avoid numerical instability (Fig. 5). For σ , the range of values are based on the results in [16]. It is observed that for both simulated and real-world cases, the proposed approach enables relatively accurate EDFs at all hyperparameter levels while providing a comparable gradient accuracy.

Lastly, for collision probabilities, the experiment from the 2D evaluation (Fig. 6) is conducted for the given 3D point clouds (Fig. 9). A spherical covariance of $\sigma_p^2 \mathbf{I}_3$ is used for the robot position uncertainty. As expected, for both simulated and real-world point clouds the collision probabilities decrease as the distance from the surfaces increases. Moreover, it is observed that in the occluded regions (i.e., regions behind the surface when viewed along the green arrows in Fig. 9) the blended approach does not provide reliable estimates. This is because the dot product between the surface normals and distance vectors is negative. It is reasonable to expect the estimates in the occluded regions to be degraded until the occluded regions are observed. The unblended approach provides estimates without this consideration which may be risky during navigation. The blending approach results in smoother probability isocontour lines (lower noise in dashed orange lines) in the visible of the 3D experiments, also, making the approach valuable for continuous-space queries in future 3D uncertainty-aware motion planning frameworks.

V. CONCLUSION

This work detailed collision probability, Euclidean distance and gradient estimation for an ellipsoidal robot from a surface that is represented as a set of ellipsoids derived from Gaussian distributions. Prior work in ellipsoid-to-ellipsoid distance estimation was extended to compute distance and gradient in the proposed context. A geometrical blending approach ensured that the estimated collision probabilities are smooth so that

they can be used for uncertainty-aware motion planning. These methods were validated using 2D and 3D real-world point cloud environments, demonstrating superior performance (as much as $10\times$ in the 2D case) compared to the state-of-the-art continuous space method. Lastly, the computational performance of the proposed methods indicates that the distance, gradient, and collision probability estimation can be executed in a single-thread on low-power CPUs of embedded computers (e.g. NVIDIA Orin) in a few milliseconds for thousands of ellipsoids.

There are two key limitations of this work. First, the computation in Eq. (6) may require additional local submap extraction or spatial partitioning data structures to enable scalability as M increases. A concurrent or vectorized implementation of the eigenvalue problems in Proposition 1 may further improve performance. Second, estimation in the orientation space (formally, in the special orthogonal group $\text{SO}(2)$ or $\text{SO}(3)$) is not explicitly considered in this work which may be an interesting direction for future research.

ACKNOWLEDGMENTS

This work was supported in part by an Uber Presidential Fellowship. The authors thank J. Lee, M. Hansen, and D. Wetergreen for feedback on this manuscript.

REFERENCES

- [1] Ali-akbar Agha-mohammadi, Eric Heiden, Karol Hausman, and Gaurav Sukhatme. Confidence-rich grid mapping. *The International Journal of Robotics Research*, 38(12-13):1352–1374, October 2019. ISSN 0278-3649. doi: 10.1177/0278364919839762.
- [2] Christopher M. Bishop. Pattern recognition and machine learning. Information Science and Statistics. Springer, New York, 2006. ISBN 978-0-387-31073-2.
- [3] Adam Bry and Nicholas Roy. Rapidly-exploring Random Belief Trees for motion planning under uncertainty. In *2011 IEEE International Conference on Robotics and Automation*, pages 723–730, Shanghai, China, May 2011. IEEE. ISBN 978-1-61284-386-5. doi: 10.1109/ICRA.2011.5980508.
- [4] Yizhou Chen, Shupeng Lai, Jinqiang Cui, Biao Wang, and Ben M. Chen. GPU-Accelerated Incremental Euclidean Distance Transform for Online Motion Planning of Mobile Robots. *IEEE Robotics and Automation Letters*, 7(3):6894–6901, July 2022. ISSN 2377-3766. doi: 10.1109/LRA.2022.3177852.
- [5] Sungjoon Choi, Qian-Yi Zhou, and Vladlen Koltun. Robust Reconstruction of Indoor Scenes. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5556–5565, 2015.
- [6] Keenan Crane, Clarisse Weischedel, and Max Wardetzky. The heat method for distance computation. *Communications of the ACM*, 60(11):90–99, October 2017. ISSN 0001-0782, 1557-7317. doi: 10.1145/3131280.
- [7] Aditya Dhawale and Nathan Michael. Efficient Parametric Multi-Fidelity Surface Mapping. In *Robotics: Science and Systems XVI*. Robotics: Science and Systems Foundation, July 2020. ISBN 978-0-9923747-6-1. doi: 10.15607/RSS.2020.XVI.073.
- [8] Aditya Dhawale, Xuning Yang, and Nathan Michael. Reactive Collision Avoidance Using Real-Time Local Gaussian Mixture Model Maps. In *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 3545–3550, October 2018. doi: 10.1109/IROS.2018.8593723.
- [9] Noel E. Du Toit and J. W. Burdick. Probabilistic Collision Checking With Chance Constraints. *IEEE Transactions on Robotics*, 27(4):809–815, August 2011. ISSN 1552-3098, 1941-0468. doi: 10.1109/TRO.2011.2116190.
- [10] Ben Eckart, Kihwan Kim, Alejandro Troccoli, Alonzo Kelly, and Jan Kautz. Accelerated Generative Models for 3D Point Cloud Data. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 5497–5505, Las Vegas, NV, USA, June 2016. IEEE. ISBN 978-1-4673-8851-1. doi: 10.1109/CVPR.2016.593.
- [11] A. Elfes. Using occupancy grids for mobile robot perception and navigation. *Computer*, 22(6):46–57, June 1989. ISSN 1558-0814. doi: 10.1109/2.30720.
- [12] Peter R. Florence, John Carter, Jake Ware, and Russ Tedrake. NanoMap: Fast, Uncertainty-Aware Proximity Queries with Lazy Search Over Local 3D Data. In *2018 IEEE International Conference on Robotics and Automation (ICRA)*, pages 7631–7638, May 2018. doi: 10.1109/ICRA.2018.8463195.
- [13] Gilad Francis, Lionel Ott, and Fabio Ramos. Functional Path Optimisation for Exploration in Continuous Occupancy Maps. In Nancy M. Amato, Greg Hager, Shawna Thomas, and Miguel Torres-Torriti, editors, *Robotics Research*, Springer Proceedings in Advanced Robotics, pages 859–875, Cham, 2020. Springer International Publishing. ISBN 978-3-030-28619-4. doi: 10.1007/978-3-030-28619-4_59.
- [14] Yarin Gal and Zoubin Ghahramani. Dropout as a Bayesian Approximation: Representing Model Uncertainty in Deep Learning. In *Proceedings of The 33rd International Conference on Machine Learning*, pages 1050–1059. PMLR, June 2016.
- [15] Kshitij Goel and Wennie Tabib. Incremental Multimodal Surface Mapping via Self-Organizing Gaussian Mixture Models. *IEEE Robotics and Automation Letters*, 8(12):8358–8365, December 2023. ISSN 2377-3766, 2377-3774. doi: 10.1109/LRA.2023.3327670.
- [16] Kshitij Goel, Nathan Michael, and Wennie Tabib. Probabilistic Point Cloud Modeling via Self-Organizing Gaussian Mixture Models. *IEEE Robotics and Automation Letters*, 8(5):2526–2533, May 2023. ISSN 2377-3766. doi: 10.1109/LRA.2023.3256923.
- [17] Gaël Guennebaud, Benoît Jacob, et al. Eigen v3. <http://eigen.tuxfamily.org>, 2010.
- [18] Abhishek Halder. On the Parameterized Computation of Minimum Volume Outer Ellipsoid of Minkowski Sum of Ellipsoids. In *2018 IEEE Conference on Decision and Control (CDC)*, pages 4040–4045, Miami Beach, FL, December 2018. IEEE. ISBN 978-1-5386-1395-5. doi: 10.1109/CDC.2018.8619508.
- [19] Luxin Han, Fei Gao, Boyu Zhou, and Shaojie Shen. FIESTA: Fast Incremental Euclidean Distance Fields for Online Motion Planning of Aerial Robots. In *2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 4423–4430, November 2019. doi: 10.1109/IROS40897.2019.8968199.
- [20] Eric Heiden, Karol Hausman, Gaurav S. Sukhatme, and Ali-akbar Agha-mohammadi. Planning high-speed safe trajectories in confidence-rich maps. In *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 2880–2886, September 2017. doi: 10.1109/IROS.2017.8206120.
- [21] Armin Hornung, Kai M. Wurm, Maren Bennewitz, Cyrill Stachniss, and Wolfram Burgard. OctoMap: An efficient probabilistic 3D mapping framework based on octrees. *Autonomous Robots*, 34(3):189–206, April 2013. ISSN 1573-7527. doi: 10.1007/s10514-012-9321-0.
- [22] Jeffrey Ichnowski and Ron Alterovitz. Concurrent Nearest-Neighbor Searching for Parallel Sampling-Based Motion Planning in $\text{SO}(3)$, $\text{SE}(3)$, and Euclidean Spaces. In Marco Morales, Lydia Tapia, Gildardo Sánchez-Ante, and Seth Hutchinson, editors, *Algorithmic Foundations of Robotics XIII*, Springer Proceedings in Advanced Robotics, pages 69–85, Cham, 2020. Springer International Publishing. ISBN 978-3-030-44051-0. doi: 10.1007/978-3-030-44051-0_5.
- [23] Hongcheng Ji, Haibo Xie, Cheng Wang, and Huayong Yang. E-RRT*: Path Planning for Hyper-Redundant Manipulators. *IEEE Robotics and Automation Letters*, 8(12):8128–8135, December 2023. ISSN 2377-3766, 2377-3774. doi: 10.1109/LRA.2023.3325716.
- [24] Simon J. Julier and Jeffrey K. Uhlmann. New extension of the Kalman filter to nonlinear systems. In *Signal Processing, Sensor Fusion, and Target Recognition VI*, volume 3068, pages 182–193. SPIE, July 1997. doi: 10.1117/12.280797.
- [25] Bernhard Kerbl, Georgios Kopanas, Thomas Leimkuehler, and George Drettakis. 3D Gaussian Splatting for Real-Time Radiance Field Rendering. *ACM Transactions on Graphics*, 42(4):139:1–139:14, July 2023. ISSN 0730-0301. doi: 10.1145/3592433.
- [26] Leonid Keselman and Martial Hebert. Approximate Differentiable Rendering with Algebraic Surfaces. In *Computer Vision – ECCV 2022: 17th European Conference, Tel Aviv, Israel, October 23–27, 2022, Proceedings, Part XXXII*, pages 596–614, Berlin, Heidelberg, October 2022. Springer-Verlag. ISBN 978-3-031-19823-6. doi: 10.1007/978-3-031-19824-3_35.

- [27] Johann Laconte, Christophe Debain, Roland Chapuis, François Pomerleau, and Romuald Aufrère. Lambda-Field: A Continuous Counterpart of the Bayesian Occupancy Grid for Risk Assessment. In *2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 167–172, November 2019. doi: 10.1109/IROS40897.2019.8968100.
- [28] Jean-Claude Latombe. Robot Motion Planning. Springer US, Boston, MA, 1991. ISBN 978-0-7923-9206-4 978-1-4615-4022-9. doi: 10.1007/978-1-4615-4022-9.
- [29] Cedric Le Gentil, Othmane-Latif Ouabi, Lan Wu, Cedric Pradalier, and Teresa Vidal-Calleja. Accurate Gaussian-Process-Based Distance Fields With Applications to Echolocation and Mapping. *IEEE Robotics and Automation Letters*, 9(2):1365–1372, February 2024. ISSN 2377-3766. doi: 10.1109/LRA.2023.3346759.
- [30] Borham Lee, Clark Zhang, Zonghao Huang, and Daniel D. Lee. Online Continuous Mapping using Gaussian Process Implicit Surfaces. In *2019 International Conference on Robotics and Automation (ICRA)*, pages 6884–6890, May 2019. doi: 10.1109/ICRA.2019.8794324.
- [31] Peter Zhi Xuan Li, Sertac Karaman, and Vivienne Sze. Memory-Efficient Gaussian Fitting for Depth Images in Real Time. In *2022 International Conference on Robotics and Automation (ICRA)*, pages 8003–8009, May 2022. doi: 10.1109/ICRA46639.2022.9811682.
- [32] Peter Zhi Xuan Li, Sertac Karaman, and Vivienne Sze. GMMMap: Memory-Efficient Continuous Occupancy Map Using Gaussian Mixture Model. *IEEE Transactions on Robotics*, 40:1339–1355, 2024. ISSN 1552-3098, 1941-0468. doi: 10.1109/TRO.2023.3348305.
- [33] Anhua Lin and Shih-Ping Han. On the Distance between Two Ellipsoids. *SIAM Journal on Optimization*, 13(1):298–308, January 2002. ISSN 1052-6234, 1095-7189. doi: 10.1137/S1052623401396510.
- [34] Sikang Liu, Kartik Mohta, Nikolay Atanasov, and Vijay Kumar. Search-Based Motion Planning for Aggressive Flight in SE(3). *IEEE Robotics and Automation Letters*, 3(3):2439–2446, July 2018. ISSN 2377-3766, 2377-3774. doi: 10.1109/LRA.2018.2795654.
- [35] Tianyu Liu, Fu Zhang, Fei Gao, and Jia Pan. Tight Collision Probability for UAV Motion Planning in Uncertain Environment. In *2023 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 1055–1062, Detroit, MI, USA, October 2023. IEEE. ISBN 978-1-66549-190-7. doi: 10.1109/IROS55552.2023.10342141.
- [36] Tomás Lozano-Pérez and Michael A. Wesley. An algorithm for planning collision-free paths among polyhedral obstacles. *Communications of the ACM*, 22(10):560–570, October 1979. ISSN 0001-0782, 1557-7317. doi: 10.1145/359156.359164.
- [37] A. M. Mathai and S. B. Provost. Quadratic Forms in Random Variables: Theory and Applications. Marcel Dekker, 1992.
- [38] Ben Mildenhall, Pratul P. Srinivasan, Matthew Tancik, Jonathan T. Barron, Ravi Ramamoorthi, and Ren Ng. NeRF: Representing Scenes as Neural Radiance Fields for View Synthesis. In Andrea Vedaldi, Horst Bischof, Thomas Brox, and Jan-Michael Frahm, editors, *Computer Vision – ECCV 2020*, Lecture Notes in Computer Science, pages 405–421, Cham, 2020. Springer International Publishing. ISBN 978-3-030-58452-8. doi: 10.1007/978-3-030-58452-8_24.
- [39] Anastasios I. Mourikis and Stergios I. Roumeliotis. A Multi-State Constraint Kalman Filter for Vision-aided Inertial Navigation. In *Proceedings 2007 IEEE International Conference on Robotics and Automation*, pages 3565–3572, Rome, Italy, April 2007. IEEE. ISBN 978-1-4244-0602-9 978-1-4244-0601-2. doi: 10.1109/ROBOT.2007.364024.
- [40] Ken Museth. VDB: High-resolution sparse volumes with dynamic topology. *ACM Transactions on Graphics*, 32(3):1–22, June 2013. ISSN 0730-0301, 1557-7368. doi: 10.1145/2487228.2487235.
- [41] Huan Nguyen, Sondre Holm Fyhn, Paolo De Petris, and Kostas Alexis. Motion Primitives-based Navigation Planning using Deep Collision Prediction. In *2022 International Conference on Robotics and Automation (ICRA)*, pages 9660–9667, Philadelphia, PA, USA, May 2022. IEEE. ISBN 978-1-72819-681-7. doi: 10.1109/ICRA46639.2022.9812231.
- [42] Helen Oleynikova, Zachary Taylor, Marius Fehr, Roland Siegwart, and Juan Nieto. Voxblox: Incremental 3D Euclidean Signed Distance Fields for on-board MAV planning. In *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 1366–1373, September 2017. doi: 10.1109/IROS.2017.8202315.
- [43] Cormac O’Meara, Wennie Tabib, and Nathan Michael. Variable Resolution Occupancy Mapping Using Gaussian Mixture Models. *IEEE Robotics and Automation Letters*, 4(2):2015–2022, April 2019. ISSN 2377-3766. doi: 10.1109/LRA.2018.2889348.
- [44] Joseph Ortiz, Alexander Clegg, Jing Dong, Edgar Sucar, David Novotny, Michael Zollhoefer, and Mustafa Mukadam. iSDF: Real-Time Neural Signed Distance Fields for Robot Perception. In *Robotics: Science and Systems XVIII*, volume 18, June 2022. ISBN 978-0-9923747-8-5.
- [45] Fabio Ramos and Lionel Ott. Hilbert maps: Scalable continuous occupancy mapping with stochastic gradient descent. *The International Journal of Robotics Research*, 35(14):1717–1730, December 2016. ISSN 0278-3649, 1741-3176. doi: 10.1177/0278364916684382.
- [46] Carl Edward Rasmussen and Christopher K. I. Williams. Gaussian Processes for Machine Learning. The MIT Press, November 2005. ISBN 978-0-262-25683-4. doi: 10.7551/mitpress/3206.001.0001.
- [47] Nathan Ratliff, Matt Zucker, J. Andrew Bagnell, and Siddhartha Srinivasa. CHOMP: Gradient optimization techniques for efficient motion planning. In *2009 IEEE International Conference on Robotics and Automation*, pages 489–494, May 2009. doi: 10.1109/ROBOT.2009.5152817.
- [48] Elon Rimon and Stephen P. Boyd. Obstacle Collision Detection Using Best Ellipsoid Fit. *Journal of Intelligent and Robotic Systems*, 18(2):105–126, February 1997. ISSN 1573-0409. doi: 10.1023/A:1007960531949.
- [49] Sipu Ruan, Karen L. Poblete, Hongtao Wu, Qianli Ma, and Gregory S. Chirikjian. Efficient Path Planning in Narrow Passages for Robots With Ellipsoidal Components. *IEEE Transactions on Robotics*, 39(1):110–127, February 2023. ISSN 1941-0468. doi: 10.1109/TRO.2022.3187818.
- [50] Shobhit Srivastava and Nathan Michael. Efficient, Multifidelity Perceptual Representations via Hierarchical Gaussian Mixture Models. *IEEE Transactions on Robotics*, 35(1):248–260, February 2019. ISSN 1941-0468. doi: 10.1109/TRO.2018.2878363.
- [51] G. W. Stewart. The Efficient Generation of Random Orthogonal Matrices with an Application to Condition Estimators. *SIAM Journal on Numerical Analysis*, 17(3):403–409, 1980. ISSN 0036-1429.
- [52] Wennie Tabib, Kshitij Goel, John Yao, Mosam Dabhi, Curtis Boirum, and Nathan Michael. Real-Time Information-Theoretic Exploration with Gaussian Mixture Model Maps. In *Robotics: Science and Systems XV*. Robotics: Science and Systems Foundation, June 2019. ISBN 978-0-9923747-5-4. doi: 10.15607/RSS.2019.XV.061.
- [53] Antony Thomas, Fulvio Mastrogiovanni, and Marco Baglietto. Exact and Bounded Collision Probability for Motion Planning Under Gaussian Uncertainty. *IEEE Robotics and Automation Letters*, 7(1):167–174, January 2022. ISSN 2377-3766, 2377-3774. doi: 10.1109/LRA.2021.3121073.
- [54] Jesus Tordesillas, Brett T. Lopez, Michael Everett, and Jonathan P. How. FASTER: Fast and Safe Trajectory Planner for Navigation in Unknown Environments. *IEEE Transactions on Robotics*, 38(2):922–938, April 2022. ISSN 1552-3098, 1941-0468. doi: 10.1109/TRO.2021.3100142.
- [55] Kevin Tracy, Taylor A. Howell, and Zachary Manchester. Differentiable Collision Detection for a Set of Convex Primitives. In *2023 IEEE International Conference on Robotics and Automation (ICRA)*, pages 3663–3670, London, United Kingdom, May 2023. IEEE. ISBN 9798350323658. doi: 10.1109/ICRA48891.2023.10160716.
- [56] Oliver Williams and Andrew Fitzgibbon. Gaussian Process Implicit Surfaces. Technical report, 2007.
- [57] Lan Wu, Ki Myung Brian Lee, Liyang Liu, and Teresa Vidal-Calleja. Faithful Euclidean Distance Field From Log-Gaussian Process Implicit Surfaces. *IEEE Robotics and Automation Letters*, 6(2):2461–2468, April 2021. ISSN 2377-3766. doi: 10.1109/LRA.2021.3061356.
- [58] Lan Wu, Cedric Le Gentil, and Teresa Vidal-Calleja. Pseudo Inputs Optimisation for Efficient Gaussian Process Distance Fields. In *2023 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 7249–7255, October 2023. doi: 10.1109/IROS55552.2023.10342483.
- [59] Lan Wu, Ki Myung Brian Lee, Cedric Le Gentil, and Teresa Vidal-Calleja. Log-GPIS-MOP: A Unified Representation for Mapping, Odometry, and Planning. *IEEE Transactions on Robotics*, 39(5):4078–4094, October 2023. ISSN 1941-0468. doi: 10.1109/TRO.2023.3296982.
- [60] Qian-Yi Zhou and Vladlen Koltun. Dense scene reconstruction with points of interest. *ACM Transactions on Graphics*, 32(4):1–8, July 2013. ISSN 0730-0301, 1557-7368. doi: 10.1145/2461912.2461919.
- [61] Hai Zhu and Javier Alonso-Mora. Chance-Constrained Collision Avoidance for MAVs in Dynamic Environments. *IEEE Robotics and Automation Letters*, 4(2):776–783, April 2019. ISSN 2377-3766, 2377-3774. doi: 10.1109/LRA.2019.2893494.