

LF-3PM: a LiDAR-based Framework for Perception-aware Planning with Perturbation-induced Metric

Kaixin Chai^{†1}, Long Xu^{†1}, Qianhao Wang¹, Chao Xu¹, and Fei Gao¹

Abstract— Just as humans can become disoriented in featureless deserts or thick fogs, not all environments are conducive to the Localization Accuracy and Stability (LAS) of autonomous robots. This paper introduces an efficient framework designed to enhance LiDAR-based LAS through strategic trajectory generation, known as Perception-aware Planning. Unlike vision-based frameworks, the LiDAR-based requires different considerations due to unique sensor attributes. Our approach focuses on two main aspects: firstly, assessing the impact of LiDAR observations on LAS. We introduce a perturbation-induced metric to provide a comprehensive and reliable evaluation of LiDAR observations. Secondly, we aim to improve motion planning efficiency. By creating a Static Observation Loss Map (SOLM) as an intermediary, we logically separate the time-intensive evaluation and motion planning phases, significantly boosting the planning process. In the experimental section, we demonstrate the effectiveness of the proposed metrics across various scenes and the feature of trajectories guided by different metrics. Ultimately, our framework is tested in a real-world scenario, enabling the robot to actively choose topologies and orientations preferable for localization. The source code is accessible at <https://github.com/ZJU-FAST-Lab/LF-3PM>.

I. INTRODUCTION

Accurate localization is important for autonomous robots to perform complex tasks, especially in GPS-denied environments. Although localization algorithms [1]–[4] are reliable in most cases, scenarios still exist where Localization Accuracy and Stability (LAS) can get poor. One solution is to make the robot actively choose trajectories conducive to improving LAS, known as Perception-aware Planning.

In general, a Perception-aware Planning framework pivots on two essential components: observation evaluation and trajectory generation. By evaluating the accumulated loss of observations simulated along the candidate trajectories, the robots can choose one that is most conducive to improving LAS. Recently, considerable progress [5]–[9] has been made on vision-based Perception-aware Planning frameworks, yet related research based on LiDAR remains limited. To improve the LiDAR-based LAS of robots, it’s intuitive to adapt the design principles from vision-based frameworks. However, the essential components require significant modifications due to the distinct attributes of cameras and LiDAR.

On the one hand, the focus of observation evaluation in the framework of the two kinds is different. The main factors that affect the vision-based LAS are texture richness [5] and lighting conditions [10], while for the LiDAR-based LAS, it is the geometric structure of the surroundings [11].

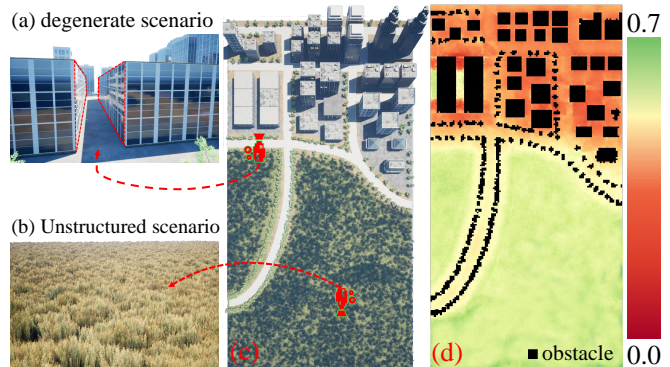


Fig. 1: (a) Narrow corridor between two buildings. (b) Lush grasslands. (c) Bird’s eye view of the entire map. (d) SOLM of the map with the observation model given by Eq. (25), where areas with smaller observation loss are preferable for localization and black areas indicate obstacles.

On the other hand, the efficiency of simulating visual feature points is much higher than that of simulating a LiDAR scan. The visual feature points are sparse enough to allow real-time simulation [5], even enabling the construction of visibility functions for each feature point [12]. However, LiDAR points are dense, making it impractical to simulate scans during motion planning.

Therefore, we need a new metric for LiDAR-based observation loss evaluation to produce reliable guidance in trajectory generation and an efficient framework that rationally separates observation evaluation from motion planning to prevent the time-consuming simulation from blocking the trajectory generation process.

We note that the LiDAR-based LAS primarily depends on the geometric structure of the environment, which is inherently more stable than vision-based LAS due to its invariance to lighting conditions. This stability permits the pre-evaluation of the loss of LiDAR observation at the robot’s potential state space like $SE(2)$ or $SE(3)$. In addition, since observations are roughly the same between two similar poses, the loss does not drastically change within a small area with a slight orientation difference. Thus, we can discretize the state space into grids, evaluate the observation loss once at the center of each grid, and then use interpolation to obtain an evaluation of the intermediate states. We name the grid structure as Static Observation Loss Map (SOLM).

As mentioned earlier, we need to tailor a new metric to evaluate LiDAR observation loss at each grid in SOLM. While covariance is a widely used metric [6] to quantify the observation loss by localization uncertainty, it does not

[†]Indicates equal contribution.

¹Huzhou Institute of Zhejiang University, Huzhou 313000, China.

E-mail: kaixinchai@outlook.com, {gaolon, fgaoaa}@zju.edu.cn

offer independent evaluation at one single location because the result depends on historical observations collected in other grids. Additionally, some metrics like pre-defining the empirical principle [13] show insights into the localization process. However, lacking theoretical support may diminish the effectiveness of the metrics across diverse scenarios and with varying LiDAR configurations.

Therefore, a theoretically well-founded metric is required to evaluate LiDAR observation loss at a single location. Given that Mainstream filtering-based [2] or optimization-based [4] localization methods are essentially solving a Least Squares Problem, our analysis starts from the same form. Inspired by the sensitivity analysis of linear Least Squares Problem [14], [15], we propose a metric by introducing perturbations into LiDAR observations.

Fig. 1 shows an example of SOLM. A ground robot with a rotating LiDAR localizes itself in the environment, as shown in Fig. 1(c). Since the yaw angle does not affect observations with a 360-degree LiDAR, we calculate SOLM in \mathbb{R}^2 space to eliminate unnecessary computations. In Sec. V, a case with a limited field of view will be presented. According to the SOLM in Fig. 1(d), degenerate corridors (Fig. 1(a)) and unstructured grasslands (Fig. 1(b)) are identified as less favorable for improving LAS, whereas places with rich structured features are deemed beneficial.

During motion planning, we consider SOLM alongside other factors, such as obstacle avoidance, trajectory smoothness, and agility, to generate trajectories favorable for localization. We apply the proposed framework in more scenarios in Sec.V to validate the effectiveness. The experiments show that benefiting from the reliable metric and well-designed framework, the robot can improve its LAS by selecting topologies and orientations with low observation loss.

In summary, this paper makes the following contributions:

- 1) We propose a novel perturbation-induced metric to evaluate LiDAR observation loss in SOLM calculation.
- 2) We design an efficient LiDAR-based Perception-aware Planning framework leveraging LiDAR attributes to decouple observation evaluation and motion planning.
- 3) We conduct extensive experiments to validate the effectiveness of the proposed framework. The entire project is open-sourced to promote further research in this field.

II. RELATED WORKS

This section will discuss existing Perception-aware Planning frameworks, focusing on the design of metrics and frameworks. We aim to highlight the advancements and identify gaps our research seeks to address.

In the realm of metric design for LiDAR-based Perception-aware Planning frameworks, empirical metrics [13] have been explored to evaluate LiDAR observation loss, offering insights into the localization process. However, these metrics are not well theoretically supported, which may lead to diminished effectiveness across various scenarios and different sensor configurations, highlighting the need for more universally applicable metrics.

Some researchers have proposed metrics [7], [14], [16] that evaluate observation loss at a specific location by examining the observability of observations. These metrics are adept at identifying degenerate scenarios, such as open fields or tunnels. Yet, degenerate scenarios are not the sole contributors to the diminished LAS. Factors like dynamic objects [17], unstructured environments [18], and sensor noise [19] interference also play significant roles.

Cognetti et al. [20] recognized it is insufficient to focus only on observability without considering noise and use pose estimation covariance as the metric. It works in vision-based frameworks, but as mentioned in Sec.I, covariance is not an ideal metric to provide independent evaluation for SOLM calculation due to its historical observations. Without the help of SOLM, we have to simulate the LiDAR scan for covariance calculation during motion planning, which will reduce the efficiency of the entire framework.

Therefore, a theoretically founded and independent metric for evaluating LiDAR observation loss is necessary. This paper proposes a novel metric via perturbation and sensitivity analysis [15]. In contrast to the approach in work [14], where perturbations are applied to additional constraints, we directly apply perturbations to the observations and analyze their effect on the solution. We find that the degeneracy factor derived in work [14] is a special case of the proposed metric.

Regarding the framework for Perception-aware Planning, we focus on how existing strategies incorporate the observation loss into motion planning. Strategy [7], [12] models observation loss as a differentiable function concerning the robot's pose. This approach assumes feature points are always visible, which may not hold in complex environments. Another strategy [6] selects the final trajectory with the lowest covariance from several candidate trajectories that meet both obstacle avoidance and kinematic requirements. Strategy [5] uses covariance as a heuristic in exploring paths with Rapid Random Trees. The above three strategies work well in vision-based frameworks, while it is impractical for LiDAR-based Perception-aware Planning, primarily because vision feature points are sparse and can be simulated in real-time, whereas LiDAR point clouds are dense and simulating LiDAR scans is time-consuming, making trajectory generation has a long time span.

We decouple the evaluation process and motion planning into a two-phase pipeline to ensure the efficiency of motion planning. At first, the proposed perturbation-induced metric is employed to calculate SOLM, which is then stored as gray-scale images. When a new trajectory needs to be generated, the motion planning module will use the stored SOLM in the front- and back-end, finally optimizing a trajectory that can balance LAS, obstacle avoidance, and agility.

III. METRIC DERIVATION

In this section, we derive a new metric to evaluate LiDAR observation loss, facilitating SOLM computation. In Sec. III-A, we state the problem to be solved and give the intuition behind our method. In Sec. III-B, we present an overview

of the derivation process to outline our methodology and put details in the appendix for better readability.

A. Problem statement and method intuition

Assume the robot's pose is denoted by \mathbf{x} , which has n dimensions. Data captured at pose \mathbf{x} by sensors can be utilized to construct error-based observations, denoted as $\mathbf{h}(\mathbf{x}) = [h_1(\mathbf{x}), h_2(\mathbf{x}), \dots, h_m(\mathbf{x})]^T$. An example of $h_j(\mathbf{x})$ is given by Eq.(25) in Sec. V, which means the distance from point j to its nearest plane. We aim to develop a metric to evaluate LiDAR observation loss that quantifies the effect of a LiDAR observation $\mathbf{h}(\mathbf{x})$ on the robot LAS. The metric can be formulated as:

$$q = Q(\mathbf{h}(\mathbf{x})), \quad (1)$$

where Q denotes the metric, and scalar q is the observation loss. Regions with low q values are more conducive to maintaining good LAS.

As mentioned in Sec. I, mainstream localization algorithms [2], [4] are essentially solving a Least Square Problem. To derive a concrete form of Q , we explore how observations $\mathbf{h}(\mathbf{x})$ influence robots' LAS in the linear Least Square Problem context. Consider the pose estimation problem as the following nonlinear Least Square Problem:

$$\mathbf{x}^* = \arg \min_{\mathbf{x}} \frac{1}{2} \sum_{j=1}^m \|h_j(\mathbf{x})\|_2^2. \quad (2)$$

Usually, we solve Eq.(2) iteratively with \mathbf{x}_0 , an initial estimate of \mathbf{x} . Suppose we iterate one step using Gauss-Newton method [21] to obtain \mathbf{x}_{opt} , which is equivalent to solving the following optimization problem:

$$\mathbf{x}_{\text{opt}} = \arg \min_{\mathbf{x}} \frac{1}{2} \|A(\mathbf{x} - \mathbf{x}_0) - \mathbf{b}\|_2^2, \quad (3)$$

where $A \in \mathbb{R}^{m \times n}$, $\mathbf{b} \in \mathbb{R}^m$. Each row of A is $\nabla_{\mathbf{x}}^T h_j|_{\mathbf{x}=\mathbf{x}_0}$, each row of \mathbf{b} is $-h_j(\mathbf{x}_0)$, and optimal pose estimation \mathbf{x}_{opt} is directly affected by observations $\mathbf{h}(\mathbf{x})$.

In fact, the observations usually contain sensor noise, erroneous associations, etc., which can be regarded as a perturbation applied to the exact observations. So we actually solve for $\hat{\mathbf{x}}_{\text{opt}}$ under disturbed observations $\hat{\mathbf{h}}(\mathbf{x})$. Naturally, we expect the displacement of the solution $\|\hat{\mathbf{x}}_{\text{opt}} - \mathbf{x}_{\text{opt}}\|$ caused by perturbations as small as possible.

According to the linear Least Square Problem sensitivity theory [15], given a certain disturbance on observations, the displacement of \mathbf{x}_{opt} depends on $\mathbf{h}(\mathbf{x})$. In other words, the attributes of $\mathbf{h}(\mathbf{x})$ decide the robustness of the solution. Hence, if the metric Q effectively captures the perturbation-induced displacement of \mathbf{x}_{opt} , it can serve as a reliable measurement of observation loss at the given pose \mathbf{x} .

B. Metric derivation

If we have a smooth enough perturbation mapping $\mathcal{V}_{\delta\mathbf{v}}$ and a perturbation $\delta\mathbf{v} \in \mathbb{R}^V$ that change the observations \mathbf{h} to $\hat{\mathbf{h}}$, where $\mathcal{V}_{\delta\mathbf{v}} : \mathbf{h}(\mathbf{x}) \mapsto \hat{\mathbf{h}}(\mathbf{x}, \delta\mathbf{v})$, \mathbf{x}_{opt} will be changed to $\hat{\mathbf{x}}_{\text{opt}}$. Note that the perturbation mapping $\mathcal{V}_{\delta\mathbf{v}}$ with different

expressions will lead to different changes in \mathbf{x}_{opt} . In this work, we assume that the mapping $\mathcal{V}_{\delta\mathbf{v}}$ has the form of a linear transformation with a bias, which can be expressed as the following equations:

$$\hat{\mathbf{h}}(\mathbf{x}, \delta\mathbf{v}) = (I + \delta K)\mathbf{h}(\mathbf{x}) + \delta\mathbf{t}, \quad (4)$$

$$\delta K = g(\delta\mathbf{k}) \in \mathbb{R}^{m \times m}, \quad (5)$$

$$\delta\mathbf{v} = [\delta\mathbf{k}, \delta\mathbf{t}]^T, \quad (6)$$

where $\delta\mathbf{t} \in \mathbb{R}^m$, $\mathbf{V} = m^2 + m$. The role of g is to resize the vector $\delta\mathbf{k} \in \mathbb{R}^{m^2}$ to the matrix δK in row-major order [22]. Substituting $\hat{\mathbf{h}}(\mathbf{x})$ into Problem (3), $\hat{\mathbf{x}}_{\text{opt}}$ can be obtained by solving the following optimization problem:

$$\hat{\mathbf{x}}_{\text{opt}} = \arg \min_{\mathbf{x}} \frac{1}{2} \|(I + \delta K)A(\mathbf{x} - \mathbf{x}_0) - (\mathbf{b} - \delta K\mathbf{b} - \delta\mathbf{t})\|_2^2. \quad (7)$$

Let $\Delta\mathbf{x} = \mathbf{x} - \mathbf{x}_0$, we have:

$$\mathbf{x}_{\text{opt}} = \arg \min_{\Delta\mathbf{x}} \frac{1}{2} \|A\Delta\mathbf{x} - \mathbf{b}\|_2^2 + \mathbf{x}_0, \quad (8)$$

$$\hat{\mathbf{x}}_{\text{opt}} = \arg \min_{\Delta\mathbf{x}} \frac{1}{2} \|(A + \Delta A)\Delta\mathbf{x} - (\mathbf{b} - \Delta\mathbf{b})\|_2^2 + \mathbf{x}_0, \quad (9)$$

where $\Delta A = g(\delta\mathbf{k})A$ and $\Delta\mathbf{b} = g(\delta\mathbf{k})\mathbf{b} + \delta\mathbf{t}$.

What we care is how much small $\delta\mathbf{k}$ and $\delta\mathbf{t}$ can affect the displacement norm of the solution $\|\hat{\mathbf{x}}_{\text{opt}} - \mathbf{x}_{\text{opt}}\|$. We find that this problem can be converted to a sensitivity analysis problem for the optimal solution of a linear Least Square Problem. Work [15] summarizes a number of methods for obtaining the upper bound on the variation of the solution caused by ΔA and $\Delta\mathbf{b}$. Let the expression for the upper bound be $E(\|\Delta A\|, \|\Delta\mathbf{b}\|)$, we have

$$\|\hat{\mathbf{x}}_{\text{opt}} - \mathbf{x}_{\text{opt}}\| \leq E(\|\Delta A\|, \|\Delta\mathbf{b}\|). \quad (10)$$

Writing the perturbations $\delta\mathbf{k}$ and $\delta\mathbf{t}$ as: $\delta\mathbf{k} = r_{\mathbf{k}}\boldsymbol{\alpha}$, $\delta\mathbf{t} = r_{\mathbf{t}}\boldsymbol{\beta}$, where $\boldsymbol{\alpha}^T\boldsymbol{\alpha} = \boldsymbol{\beta}^T\boldsymbol{\beta} = 1$, $r_{\mathbf{k}} \geq 0$, $r_{\mathbf{t}} \geq 0$, and substituting them into the upper bound E , we have:

$$\begin{aligned} E(\|\Delta A\|, \|\Delta\mathbf{b}\|) &= E(\|g(\delta\mathbf{k})A\|, \|g(\delta\mathbf{k})\mathbf{b} + \delta\mathbf{t}\|) \\ &\triangleq F(r_{\mathbf{k}}\boldsymbol{\alpha}, r_{\mathbf{t}}\boldsymbol{\beta}). \end{aligned} \quad (11)$$

Treating $\boldsymbol{\alpha}, \boldsymbol{\beta}$ as constants first and linearizing F near $r_{\mathbf{k}} = r_{\mathbf{t}} = 0$ (which means the intensities of the perturbations tend to zero), we have

$$\begin{aligned} F(r_{\mathbf{k}}\boldsymbol{\alpha}, r_{\mathbf{t}}\boldsymbol{\beta}) &\approx \left. \frac{\partial F}{\partial r_{\mathbf{k}}} \right|_{r_{\mathbf{k}}=r_{\mathbf{t}}=0} r_{\mathbf{k}} + \left. \frac{\partial F}{\partial r_{\mathbf{t}}} \right|_{r_{\mathbf{k}}=r_{\mathbf{t}}=0} r_{\mathbf{t}} \\ &\triangleq d_1(\boldsymbol{\alpha}, \boldsymbol{\beta})r_{\mathbf{k}} + d_2(\boldsymbol{\alpha}, \boldsymbol{\beta})r_{\mathbf{t}}. \end{aligned} \quad (12)$$

We note that d_1 and d_2 serve as an amplifier of perturbations, which, in other words, denotes the sensitivity of F to the perturbations. As the upper bound of the displacement of the solution $\|\hat{\mathbf{x}}_{\text{opt}} - \mathbf{x}_{\text{opt}}\|$, we expect F less sensitive to the perturbations, which will constrain the sensitivity of displacement to the perturbations at a level no more than the sensitivity of F , because the inequality sign still holds for the derivation of both sides of Inequality (10) at zero. This can be easily proved by considering the functions on the left and right sides of Inequality (10) as f_1 and f_2 in

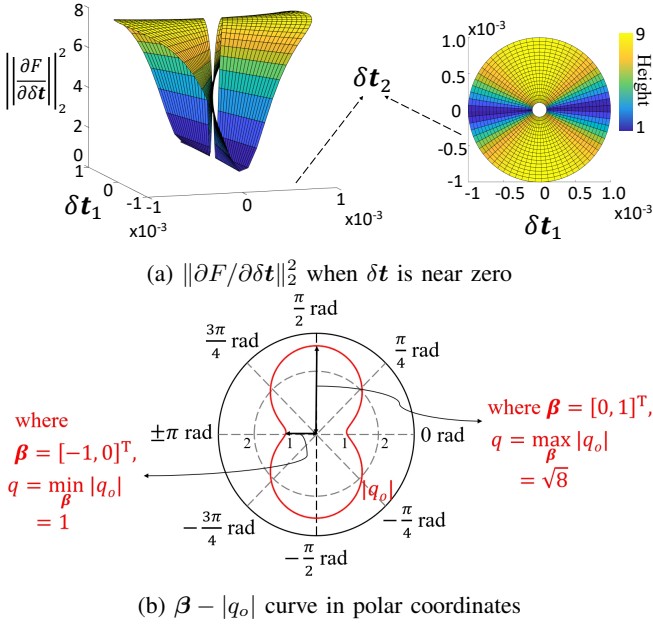


Fig. 2: A special example for illustrating the proposed metric, where $E(\|\Delta A\|, \|\Delta b\|) = \sqrt{\delta t_1^2 + 8\delta t_2^2}$, $\delta t = [\delta t_1, \delta t_2]^T \in \mathbb{R}^2$. Figure (a) illustrates the results of directly solving for F on the derivative of δt , where $\|\partial F/\partial \delta t\|_2^2 = (\delta t_1^2 + 64\delta t_2^2)/(\delta t_1^2 + 8\delta t_2^2)$. The figure to the right is a top view of the surface on the left. Since the derivative of F does not exist at zero, there is a hole in the surface. Figure (b) shows how the proposed metric looks like in the polar coordinate system $\theta - r$, where $\beta = [\cos \theta, \sin \theta]^T$, $r = |q_o| = |d_2| = \sqrt{\cos^2 \theta + 8 \sin^2 \theta}$. We can clearly see the effect of perturbations with different directions on the sensitivity of the upper bound E from this figure.

Lemma 1, respectively. Thus, we define the metric Q as the sensitivity of the upper bound F to the perturbations:

$$q = \sqrt{w_1 d_1^2(\alpha_G, \beta_G) + w_2 d_2^2(\alpha_G, \beta_G)}, \quad (13)$$

where α_G, β_G are user-chosen directions of perturbations, $w_1 > 0, w_2 > 0, w_1 + w_2 = 1$ are the weights of d_1, d_2 , respectively. Here, instead of performing sensitivity analysis on the derivation of $\|\hat{x}_{\text{opt}} - x_{\text{opt}}\|$ directly, we perform sensitivity analysis on the upper bound E because the expanded expression for $\|\hat{x}_{\text{opt}} - x_{\text{opt}}\|$ is too complicated to be analyzed. After simplifying, what we need to do is pursue observations $h(x)$ with lower q value to keep low sensitivity of displacement $\|\hat{x}_{\text{opt}} - x_{\text{opt}}\|$ to perturbations, which is conducive to improving the LAS.

The intuition for defining the metric as Eq.(13) is that the directions of the perturbations can affect the sensitivity of the upper bound E . From a mathematical point of view, it is impossible to obtain the sensitivity of E by direct derivation of δk and δt without treating α, β as constants first, since the derivatives of the norms usually do not exist at zero. To enhance the clarity of the abstract mathematical procedure, we present an example when $m = 2, w_2 = 1$, visualizing $\|\partial F/\partial \delta t\|_2^2$ and $|q_o| = \sqrt{w_1 d_1^2 + w_2 d_2^2}$ in Fig. 2.

In this work, we choose one of the upper bounds derived in the work [23] to compute the metric. According to the results of the work [23], suppose that A has full column rank and $\|\Delta A\|_2 < \sigma_1$, where σ_1 is the smallest singular value of A , we have

$$E(\|\Delta A\|, \|\Delta b\|) = \frac{\|\Delta x^*\|_2 + \sigma_1^{-1} \|r\|_2}{\sigma_1 - \|\Delta A\|_2} \|\Delta A\|_F + \frac{1}{\sigma_1 - \|\Delta A\|_2} \|\Delta b\|_2, \quad (14)$$

where $\Delta x^* = \arg \min_{\Delta x} \frac{1}{2} \|A \Delta x - b\|_2^2, r = A \Delta x^* - b$.

Writing α as $\alpha = [\delta k_1^T, \delta k_2^T, \dots, \delta k_m^T]^T$ and calculating the corresponding derivatives (Details can be found in Appendix. B), we can obtain:

$$q(\alpha) = \frac{1}{\sigma_1} \sqrt{w_1 \sum_{j=1}^m \delta k_j^T \Phi \delta k_j + w_2}, \quad (15)$$

where $\Phi = \xi^2 A A^T + b b^T, \xi = \|\Delta x^*\|_2 + \|r\|_2 / \sigma_1$.

Noting that Φ is a symmetric semi-positive definite matrix and $m \geq n$, we can diagonalize it: $\Phi = P \Lambda P^T$, where $P P^T = I, \Lambda = \text{diag}(\lambda_m, \lambda_{m-1}, \dots, \lambda_1), 0 = \lambda_1 = \lambda_2 = \dots = \lambda_{m-n-1} \leq \lambda_{m-n} \leq \dots \leq \lambda_m$ are eigenvalues of Φ . Since $\alpha^T \alpha = 1$, Eq.(15) is equivalent to:

$$q(\alpha) = \frac{1}{\sigma_1} \sqrt{w_1 \sum_{j=1}^m \delta k_j^T \Lambda \delta k_j + w_2}. \quad (16)$$

As illustrated in Fig. 2(b), the evaluation is affected by the direction of perturbations. With eigenvalue analysis, we can obtain the range of evaluation result: $\sum_{j=1}^m \delta k_j^T \Lambda \delta k_j \geq \sum_{j=1}^m \lambda_1 \delta k_j^T \delta k_j = \lambda_1$ and $\sum_{j=1}^m \delta k_j^T \Lambda \delta k_j \leq \sum_{j=1}^m \lambda_m \delta k_j^T \delta k_j = \lambda_m$.

Thus, choosing different value of $\sum_{j=1}^m \delta k_j^T \Lambda \delta k_j$, we can obtain different forms of q . The three most representative forms are:

$$q^{(\min)} = \frac{1}{\sigma_1} \sqrt{w_1 \lambda_1 + w_2} = \frac{\sqrt{w_2}}{\sigma_1}. \quad (17)$$

$$q^{(n)} = \frac{1}{\sigma_1} \sqrt{w_1 \lambda_{m-n+1} + w_2}, \quad (18)$$

$$q^{(\max)} = \frac{1}{\sigma_1} \sqrt{w_1 \lambda_m + w_2}, \quad (19)$$

Different strategies actually represent different conservativeness. $q^{(\max)}$ is the most conservative estimate, choosing the direction in which the perturbation has the greatest impact. On the contrary, $q^{(\min)}$ is the least conservative strategy, which is actually the degradation factor in literature [14]. $q^{(n)}$ is an intermediate strategy between $q^{(\max)}$ and $q^{(\min)}$, where n denotes the dimension of the state x to be estimated. In Sec. V, we will show the effect of strategies with different levels of conservativeness in observation evaluation and trajectory generation.

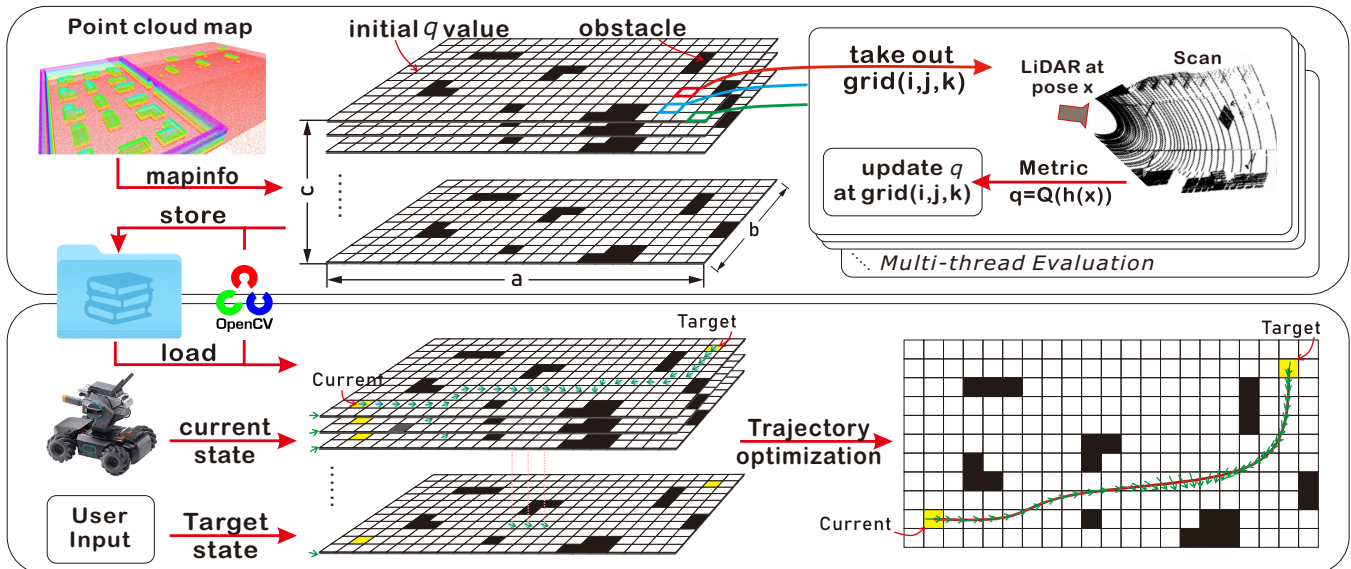


Fig. 3: The proposed perception-aware planning framework. The **top** part represents the SOLM calculation process, while the **bottom** part represents the motion planning process.

IV. FRAMEWORK DESIGN

In this section, we show the framework of the proposed LiDAR-based Perception-aware Planning, including SOLM calculation with the metrics derived in Sec. III, trajectory planning based on SOLM, and the relation of these two parts.

A. SOLM Calculation

The upper part of Fig. 3 shows the process of SOLM calculation, including initialization, updating, and saving.

To start with, we initialize the SOLM according to the robot's potential state space. Assume the robot's state is $\mathbf{x} = [x, y, \theta]^T \in SE(2)$. Given the resolution on each dimension, we discretize the $SE(2)$ state space a grid structure with the size of a, b , and c for dimensions x, y , and θ , respectively. The obstacles within the given point cloud map will be marked on the SOLM, where the observation evaluation will be skipped to reduce unnecessary calculations.

Then, we evaluate the observation loss at each grid. Firstly, the information of a non-obstacle grid (i, j, k) is taken out from the SOLM. Secondly, we simulate a LiDAR scan at the state represented by the center of this grid. A ray-casting-based scan simulator with GPU acceleration can significantly speed up the simulation process, like the one used in work [24]. The simulated scan will formulate observations $\mathbf{h}(\mathbf{x})$ follow the observation model like Eq. (25). Thirdly, we evaluate the observation loss at state \mathbf{x} with the metric Q derived in Sec. III. Finally, we update the evaluation result q into grid (i, j, k) . It is noted that since the evaluation process at different grids is independent of each other, the process can be performed in parallel. Once all the non-obstacle grids have been evaluated, we obtain the SOLM that can be used for motion planning. In practice, due to the similar structure of the picture, we store the SOLM as multi-channel images for the $SE(2)$ case. For higher-dimensional state spaces, like $SE(3)$, we store the SOLM as a tensor.

B. Motion planning

The lower part of Fig. 3 shows the process of motion planning, including path searching and trajectory optimization.

After loading SOLM from stored images, we restore the knowledge of observation loss in the robot's state space. For front-end path searching, we adopt the SOLM as the cost function and perform Breadth-First-Search (BFS) in the grid map, which allows for full consideration of all topologies and orientations in the environment to emphasize the effectiveness in improving the robot's LAS of the proposed Perception-aware Planning framework.

For trajectory optimization, similar to the work [25], we use piecewise polynomials $\mathbf{x}(t) = [x(t), y(t), \theta(t)]^T$ to represent state trajectories, where $[x, y] \in \mathbb{R}^2$ denotes the position of the robot in the world frame, $\theta \in SO(2)$ denotes the yaw angle of the robot, as mentioned in last subsection. In this work, the cubic spline is chosen to represent the trajectory for the smoothness. We formulate the perception-aware trajectory planning problem for ground robots as the following optimization problem:

$$\min_{\mathbf{c}, \mathbf{T}} \int_0^{T_s} q(\mathbf{x}(t)) dt + \rho_T T_s \quad (20)$$

$$s.t. \quad \mathbf{M}\mathbf{c} = \mathbf{b}, \quad \mathbf{T} \geq \mathbf{0}, \quad (21)$$

$$\text{Nonholo}(\mathbf{x}) = 0, \quad (22)$$

$$\dot{\mathbf{x}}^2 - ([v_{\text{mlon}}, v_{\text{mlat}}, w_{\text{max}}]^T)^2 \leq [0, 0, 0]^T, \quad (23)$$

$$\text{SDF}(\mathbf{x}) \geq r_{\text{safe}}, \quad (24)$$

where $\mathbf{c} \in \mathbb{R}^{4N \times 3}$ denote coefficient matrix, N denotes the number of segments of the piecewise polynomials. Each element in $\mathbf{T} = [T_1, T_2, \dots, T_N]^T \in \mathbb{R}^N$ denotes the duration of a piece of the trajectory. In the object function, $q(\mathbf{x}(t))$ denote observation loss at state $\mathbf{x}(t)$, $T_s = T_1 + T_2 + \dots + T_N$ is the total duration of the trajectory, ρ_T is a positive weight to ensure the aggressiveness of the trajectory.

Conditions (21) denote the continuity constraints of the cubic spline at interpolation points and the positive time, where the inequality is taken element-wise. Conditions (22) denote the possible nonholonomic constraints of the robot. For car-like robots or differential robots, it is $\dot{x} \sin \theta - \dot{y} \cos \theta = 0$. While for omnidirectional robots, this constraint does not exist. Conditions (23) are dynamic feasibility constraints, where the inequality and square operation are also taken element-wise. $v_{\text{mlon}}, v_{\text{mlat}}, w_{\text{max}}$ denote maximum longitude velocity, latitude velocity, and angular velocity, respectively. Condition (24) denotes safety constraint, where $\text{SDF}(\cdot)$ denotes the signed distance function (SDF) at position $[x, y]^T$ to the nearest obstacle. r_{safe} denotes the safety distance related to the size of the robot. In this work, we compute the SDF by the algorithm proposed in work [26].

To make the above optimization problem easier to solve, we adopt the MINCO trajectory class and differential homogeneous mapping used in work [27] to eliminate the constraints (21). For ease of solving the problem, we also replace the integral in the objective function with a summation approximation. When the equation constraint (22) exists, PHR Augmented Lagrange Multiplier method [28] (PHR-ALM) is used to solve the simplified problem for higher accuracy. Conversely, we utilize the penalty function method, which is sufficient to obtain an acceptable solution. Besides, an efficient quasi-Newton method L-BFGS [29] is chosen as the unconstrained optimization algorithm to work with PHR-ALM and the penalty function method.

V. EXPERIMENTS

A. Implementation details

To validate the effectiveness and efficiency, we apply the proposed Perception-aware Planning framework on an omnidirectional vehicle (RoboMaster AI-2020) with a 70-degree Field of View (FoV) LiDAR (Livox Mid-70¹). In real-world experiments, the odometry is provided by FAST-LIO2 [2], a high-precision Lidar-Inertial Odometry (LIO) system. The ground truth of robot pose is provided by the motion tracking system (NOKOV²). All the computation is performed on an on-board computer with Intel i7-1165G7. The observation model takes the distance from the scan point to the nearest plane, the same model defined in work [2]:

$$h_j({}^L \mathbf{p}_j, \mathbf{x}) = {}^G \mathbf{u}_j^T ({}^G \mathbf{R}_L {}^L \mathbf{p}_j + {}^G \mathbf{t}_L - \mathbf{q}_j), \quad (25)$$

where \mathbf{x} indicates the robot position ${}^G \mathbf{t}_L$ and orientation ${}^G \mathbf{R}_L$ to be estimated, ${}^L \mathbf{p}_j$ is a LiDAR point, \mathbf{u}_j is the normal vector of the nearest plane, \mathbf{q}_j is a point on the nearest plane.

To see if the metric is reliable in evaluating observation loss, we adopt an indicator Mean Disturbance-induced Error (MDE) serving as the evaluation ground truth, which is similar to e_p defined in work [30] as follows:

$$\text{MDE} = \frac{1}{N} \sum_{j=1}^N \|\log(\mathbf{T}_{\text{gt}}^{-1} \mathbf{T}_{\text{regi},j})^\vee\|_2^2, \quad (26)$$

where N denotes the number of disturbances, $\mathbf{T}_{\text{regi},j}$ is the j th registration result start from a disturbed \mathbf{T}_{gt} with iterated point-to-plane registration.

B. Metric Effectiveness Validation

In this subsection, we demonstrate the effectiveness and characteristics of the proposed metrics in evaluating observation loss in several representative scenes built in UE5³.

As shown in Fig. 4, scene (a) is rich in planar features where the robot can localize itself stably and robustly to observation perturbations. Scene (d) is a typical degenerate scenario. Scene (c) is an unstructured scenario with roughly the same features, where the robot's pose estimation is sensitive to the initial guess \mathbf{x}_0 . Scene (b) shows some structured features in the unstructured grasslands.

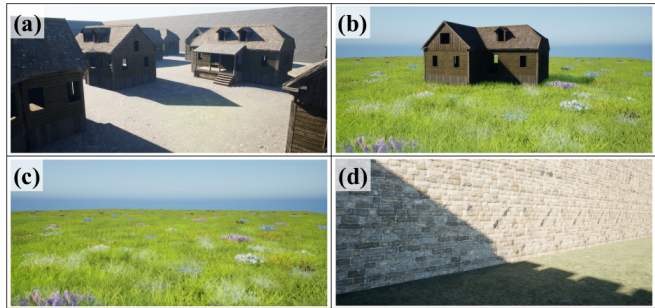


Fig. 4: Four representative scenes: (a) Houses, (b) Single house on the meadow, (c) Meadow, (d) Wall.

We first simulate a LiDAR scan at each scene to obtain observations $h(\mathbf{x})$ with the LiDAR model described in Eq. (25). Then we use metrics Eq. (17)-(19) to evaluate the loss of the simulated observation. Besides, we calculate MDE for each scene as the evaluation's ground truth. The result is shown in TABLE I. By comparing the ground truth with the evaluations by different metrics, we can judge which metric can provide the most reliable evaluation. It's important to note that we are only concerned with the relative loss of different scenarios by the same metric.

TABLE I: observation loss in four scenes

Metric	<i>a</i>	<i>b</i>	<i>c</i>	<i>d</i>
$q^{(\min)}$	0.320	0.28	0.31	12.32
$q^{(n)}$	0.547	0.91	1.20	12.34
$q^{(\max)}$	12.15	17.2	16.3	568.8
MDE	0.733	2.81	3.80	4.63

In Table I, all of the three metrics can tell the observation loss in scene (d) is the worst, while only the evaluation with metric $q^{(n)}$ is consistent with the ground truth in the four scenes. For $q^{(\min)}$, which is actually the degeneracy factor proposed in literature [14], cannot distinguish scenarios (a), (b), and (c) well because this metric only focuses on the structural richness of scenarios. In addition, the reason why the evaluation with $q^{(\max)}$ does not distinguish scenarios (b)

¹<https://www.livoxtech.com/mid-70>

²<https://www.nokov.com/>

³<https://www.ue5.com>

and (c) well is the introduction of over-conservative maximum eigenvalues, which leads to the evaluation influenced by the best-observed dimension of the state to be estimated.

C. Experiment in Large Scale Map

To demonstrate the effectiveness of the proposed framework in scenarios with large ranges, we build a Gobi desert in UE5 as shown in Fig. 5. We require the robot with the configuration in Sec. V-A to reach a distant warehouse from the inside of a small town. Stones are evenly distributed on the ground outside of towns to help robots locate themselves while traveling, and part of the stones are surrounded by weeds and bushes. To further illustrate the characteristics of the framework with different metrics, we show two trajectories guided by metrics $q^{(\min)}$ and $q^{(n)}$ respectively.

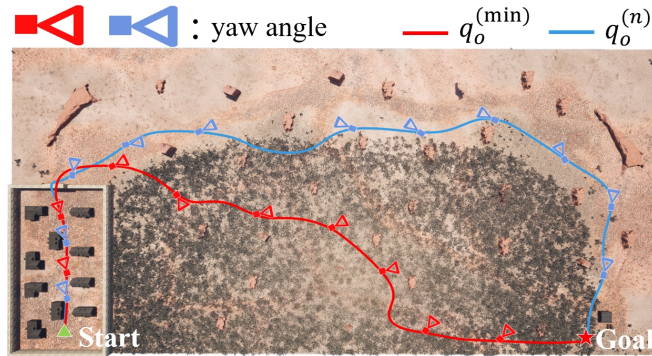


Fig. 5: Robot trajectories by the proposed framework.

As we can see in Fig. 5, the planner utilizing $q^{(\min)}$ chooses a trajectory with grass and bush, whereas the planner utilizing $q^{(n)}$ prefers the path with well-structured features and devoid of grass and bush, which is because $q^{(\min)}$ only considers the feature richness of observations and make the robot pursuit area with rich features no matter whether it is preferable for robot localization.

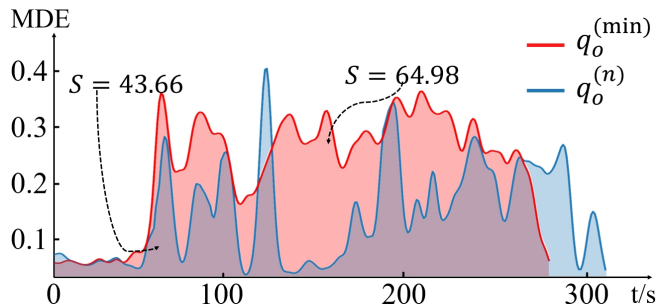


Fig. 6: MDE along two different trajectories.

To evaluate how much effect the observations along the trajectories have on LAS, we calculate MDE for each pose sampled along the trajectories at a constant time interval. As shown in Fig. 6, while the trajectory guided by $q^{(n)}$ is less rich in features compared to $q^{(\min)}$, the clear geometrical features result in smaller accumulated MDE ($S = 43.66$) compared to the trajectory guided by $q^{(\min)}$ ($S = 64.98$).

D. Real-world Experiment

After showing the different features of the proposed metrics, we adapt our Perception-aware Planning framework to

the real-world scenario. As shown in Fig. 7(a), a vehicle with configurations in Sec. V-A is required to navigate from point A to point B or C, during which process it has to localize itself with boxes and floor within the motion capture area. According to Sec. V-B and V-C, metric $q^{(n)}$ has more holistic performance that provides reliable evaluations. Here, we show trajectories (#2 and #4) guided by metric $q^{(n)}$ and trajectories (#1 and #3) without metrics for comparison.

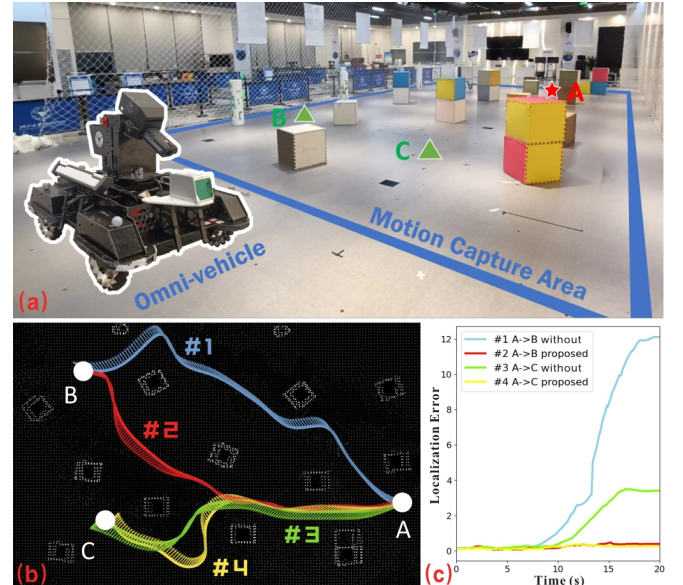


Fig. 7: (a) real-world scenario, (b) generated trajectories, (c) Localization error along different trajectories.

As shown in Fig. 7(b), the vehicle guided by $q^{(n)}$ is able to choose topologies and orientations with low observation loss. We calculate the localization error of the robot when moving along each trajectory by $\|\log(\mathbf{T}_{\text{gt}}^{-1}\mathbf{T}_{\text{odom}})^{\vee}\|_2$, where \mathbf{T}_{gt} is given by Motion Tracking System and \mathbf{T}_{odom} is given by FAST-LIO2 [2]. As we can see in Fig. 7(c), trajectories #2 and #4 have lower localization error than #1 and #3, which demonstrates that the trajectories generated by the proposed framework are conducive to improving LAS.

VI. CONCLUSION

In this paper, we propose a LiDAR-based Perception-aware Planning framework and perturbation-induced metrics. Through experiments, we show different features of the proposed metrics and validate the effectiveness of our framework. At the same time, there is still space for improvement in our work. Firstly, we assume that the localization algorithm essentially addresses a least square problem. As a result, the proposed metrics may not be suitable for all localization algorithms, such as learning-based methods. In addition, extensive real-world experiments across a broader range of scenarios are necessary to explore the limits of the proposed framework. In the future, we will conduct more experiments to investigate the effectiveness of the framework and evolve the current SOLM into a dynamically updated grid structure for online replanning applications.

APPENDIX

A. Lemma for Formula (10) ~ (13)

Lemma 1. $\forall x \in [0, +\infty)$, $f_1(x), f_2(x)$ are continuously differentiable, and $f_1(x) \leq f_2(x)$. If $f_1(0) = f_2(0)$, we have $f_1'(0) \leq f_2'(0)$.

Proof. Let $f(x) = f_2(x) - f_1(x)$, then $\forall \zeta > 0$,

$$0 \leq \frac{f_2(\zeta) - f_1(\zeta)}{\zeta} = \frac{f(\zeta) - f(0)}{\zeta}.$$

Thus,

$$0 \leq \lim_{\zeta \rightarrow 0^+} \frac{f(\zeta) - f(0)}{\zeta} = f'(0) = f_2'(0) - f_1'(0).$$

B. Derivations of Eq.(14)

$$\begin{aligned} \left. \frac{d\|\Delta A\|_F}{dr_{\mathbf{k}}} \right|_{r_{\mathbf{k}}=0} &= \left. \frac{d\sqrt{\sum_{j=1}^m \|r_{\mathbf{k}} A^T \delta \mathbf{k}_j\|_2^2}}{dr_{\mathbf{k}}} \right|_{r_{\mathbf{k}}=0} \\ &= \lim_{r_{\mathbf{k}} \rightarrow 0^+} \sqrt{\frac{r_{\mathbf{k}}^2 \sum_{j=1}^m \delta \mathbf{k}_j^T A A^T \delta \mathbf{k}_j}{r_{\mathbf{k}}^2}} \\ &= \sqrt{\sum_{j=1}^m \delta \mathbf{k}_j^T A A^T \delta \mathbf{k}_j}, \end{aligned} \quad (27)$$

$$\begin{aligned} \left. \frac{\partial \|\Delta \mathbf{b}\|_2}{\partial r_{\mathbf{t}}} \right|_{r_{\mathbf{k}}=r_{\mathbf{t}}=0} &= \left. \frac{\partial \|g(\delta \mathbf{k})\mathbf{b} + \delta \mathbf{t}\|_2}{\partial r_{\mathbf{t}}} \right|_{r_{\mathbf{k}}=r_{\mathbf{t}}=0} \\ &= \lim_{r_{\mathbf{t}} \rightarrow 0^+} \sqrt{\frac{r_{\mathbf{t}}^2 \boldsymbol{\beta}^T \boldsymbol{\beta}}{r_{\mathbf{t}}^2}} = 1, \end{aligned} \quad (28)$$

$$\begin{aligned} \left. \frac{\partial \|\Delta \mathbf{b}\|_2}{\partial r_{\mathbf{k}}} \right|_{r_{\mathbf{k}}=r_{\mathbf{t}}=0} &= \lim_{r_{\mathbf{k}} \rightarrow 0^+} \sqrt{\frac{\sum_{j=1}^m \|r_{\mathbf{k}} \mathbf{b}^T \delta \mathbf{k}_j\|_2^2}{r_{\mathbf{k}}^2}} \\ &= \sqrt{\sum_{j=1}^m \delta \mathbf{k}_j^T \mathbf{b} \mathbf{b}^T \delta \mathbf{k}_j}, \end{aligned} \quad (29)$$

REFERENCES

- [1] P. Geneva, K. Eickenhoff, W. Lee, Y. Yang, and G. Huang, "Opencvins: A research platform for visual-inertial estimation," in *2020 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2020, pp. 4666–4672.
- [2] W. Xu, Y. Cai, D. He, J. Lin, and F. Zhang, "Fast-lid2: Fast direct lidar-inertial odometry," *IEEE Transactions on Robotics*, vol. 38, no. 4, pp. 2053–2073, 2022.
- [3] T. Qin, P. Li, and S. Shen, "Vins-mono: A robust and versatile monocular visual-inertial state estimator," *IEEE Transactions on Robotics*, vol. 34, no. 4, pp. 1004–1020, 2018.
- [4] T. Shan, B. Englot, D. Meyers, W. Wang, C. Ratti, and D. Rus, "Lio-sam: Tightly-coupled lidar inertial odometry via smoothing and mapping," in *2020 IEEE/RSJ international conference on intelligent robots and systems (IROS)*. IEEE, 2020, pp. 5135–5142.
- [5] G. Costante, C. Forster, J. Delmerico, P. Valigi, and D. Scaramuzza, "Perception-aware path planning," *IEEE Transactions on Robotics*, pp. Epub-ahead, 2016.
- [6] Z. Zhang and D. Scaramuzza, "Perception-aware receding horizon navigation for mavs," in *2018 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2018, pp. 2534–2541.
- [7] —, "Beyond point clouds: Fisher information field for active visual localization," in *2019 International Conference on Robotics and Automation (ICRA)*. IEEE, 2019, pp. 5986–5992.
- [8] L. Bartolomei, L. Teixeira, and M. Chli, "Perception-aware path planning for uavs using semantic segmentation," in *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2020, pp. 5808–5815.
- [9] X. Chen, Y. Zhang, B. Zhou, and S. Shen, "Apacex: Agile and perception-aware trajectory generation for quadrotor flights," *arXiv preprint arXiv:2403.08365*, 2024.
- [10] D. Falanga, P. Foehn, P. Lu, and D. Scaramuzza, "Pampc: Perception-aware model predictive control for quadrotors," in *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2018, pp. 1–8.
- [11] J. Zhang and S. Singh, "Loam: Lidar odometry and mapping in real-time," in *Robotics: Science and systems*, vol. 2, no. 9. Berkeley, CA, 2014, pp. 1–9.
- [12] Z. Zhang and D. Scaramuzza, "Fisher information field: an efficient and differentiable map for perception-aware planning," *arXiv preprint arXiv:2008.03324*, 2020.
- [13] R. Takemura and G. Ishigami, "Perception-aware receding horizon path planning for uavs with lidar-based slam," in *2022 IEEE International Conference on Multisensor Fusion and Integration for Intelligent Systems (MFI)*. IEEE, 2022, pp. 1–8.
- [14] J. Zhang, M. Kaess, and S. Singh, "On degeneracy of optimization-based state estimation problems," in *2016 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2016, pp. 809–816.
- [15] J. F. Grcar, "Optimal sensitivity analysis of linear least squares," *Lawrence Berkeley National Laboratory, Report LBNL-52434*, vol. 99, pp. 27–34, 2003.
- [16] P. Salaris, R. Spica, P. R. Giordano, and P. Rives, "Online optimal active sensing control," in *2017 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2017, pp. 672–678.
- [17] W. Ding, S. Hou, H. Gao, G. Wan, and S. Song, "Lidar inertial odometry aided robust lidar localization system in changing city scenes," in *2020 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2020, pp. 4322–4328.
- [18] J. Guivant, E. Nebot, J. Nieto, and F. Masson, "Navigation and mapping in large unstructured environments," *The International Journal of Robotics Research*, vol. 23, no. 4-5, pp. 449–472, 2004.
- [19] C. Yuan, W. Xu, X. Liu, X. Hong, and F. Zhang, "Efficient and probabilistic adaptive voxel mapping for accurate online lidar odometry," *IEEE Robotics and Automation Letters*, vol. 7, no. 3, pp. 8518–8525, 2022.
- [20] M. Cognetti, P. Salaris, and P. R. Giordano, "Optimal active sensing with process and measurement noise," in *2018 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2018, pp. 2118–2125.
- [21] S. J. Wright, *Numerical optimization*. Springer, pp. 254–257, 2006.
- [22] D. E. Knuth, *The Art of Computer Programming: Fundamental Algorithms, volume 1*. Addison-Wesley Professional, pp. 298–305, 1997.
- [23] C. L. Lawson and R. J. Hanson, *Solving least squares problems*. SIAM, pp. 41–52, 1995.
- [24] F. Kong, X. Liu, B. Tang, J. Lin, Y. Ren, Y. Cai, F. Zhu, N. Chen, and F. Zhang, "Marsim: A light-weight point-realistic simulator for lidar-based uavs," *IEEE Robotics and Automation Letters*, vol. 8, no. 5, pp. 2954–2961, 2023.
- [25] L. Xu, K. Chai, Z. Han, H. Liu, C. Xu, Y. Cao, and F. Gao, "An efficient trajectory planner for car-like robots on uneven terrain," in *2023 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2023, pp. 2853–2860.
- [26] P. F. Felzenszwalb and D. P. Huttenlocher, "Distance transforms of sampled functions," *Theory of computing*, vol. 8, no. 1, pp. 415–428, 2012.
- [27] Z. Wang, X. Zhou, C. Xu, and F. Gao, "Geometrically constrained trajectory optimization for multicopters," *IEEE Transactions on Robotics*, vol. 38, no. 5, pp. 3259–3278, 2022.
- [28] R. T. Rockafellar, "Augmented lagrange multiplier functions and duality in nonconvex programming," *SIAM Journal on Control*, vol. 12, no. 2, pp. 268–285, 1974.
- [29] D. C. Liu and J. Nocedal, "On the limited memory bfgs method for large scale optimization," *Mathematical programming*, vol. 45, no. 1, pp. 503–528, 1989.
- [30] J. Nubert, E. Walther, S. Khattak, and M. Hutter, "Learning-based localization for robust lidar localization," in *2022 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2022, pp. 17–24.