

Adaptive Planning Framework for UAV-Based Surface Inspection in Partially Unknown Indoor Environments

Hanyu Jin, Zhefan Xu, Haoyu Shen, Xinming Han, Kanlong Ye, and Kenji Shimada

Abstract—Inspecting indoor environments such as tunnels, industrial facilities, and construction sites is essential for infrastructure monitoring and maintenance. While manual inspection in these environments is often time-consuming and potentially hazardous, Unmanned Aerial Vehicles (UAVs) can improve efficiency by autonomously handling inspection tasks. Such inspection tasks usually rely on reference maps for coverage planning. However, in industrial applications, only the floor plans are typically available. The unforeseen obstacles not included in the floor plans will result in outdated reference maps and inefficient or unsafe inspection trajectories. In this work, we propose an adaptive inspection framework that integrates global coverage planning with local reactive adaptation to improve the coverage and efficiency of UAV-based inspection in partially unknown indoor environments. Experimental results in structured indoor scenarios demonstrate the effectiveness of the proposed approach in inspection efficiency and achieving high coverage rates with adaptive obstacle handling, highlighting its potential for enhancing the efficiency of indoor facility inspection.

I. INTRODUCTION

Indoor inspection has been a critical problem for infrastructure maintenance. While manual inspection is often hazardous or inefficient, Unmanned Aerial Vehicles (UAVs) have become increasingly popular in such scenes due to their agility and potential to navigate complex environments. To achieve high-quality inspection and reconstruction, UAVs must plan and execute trajectories that provide complete and consistent coverage of target surfaces, even with environmental complexity and operational constraints.

Existing UAV inspection methods typically rely on reference maps that define the inspection target [1][2][3], which can be utilized to plan global coverage trajectories that aim to minimize flight time while ensuring complete surface visibility. However, these prior maps can be outdated or incomplete due to the unmodeled static or dynamic obstacles such as temporary structures, pieces of equipment, or human workers. Such discrepancies may not only introduce navigational challenges but also result in occluded or unreachable viewpoints, leading to coverage gaps and degraded reconstruction quality.

To address these limitations, this work proposes a hierarchical planning framework for partially unknown indoor environments. Based on our previous work on tunnel inspection [4], our system integrates a segment-based global viewpoint planner for coverage planning and a local planner that adapts



Fig. 1. A UAV performing autonomous inspection in a tunnel-like indoor environment using the proposed method.

the UAV trajectory and viewing angle in real time. This integration enables the UAV to preserve inspection quality in environments with unforeseen obstacles. Figure 1 shows the deployment of our framework in tunnel wall inspection.

The main contributions of this work include:

- **Surface Inspection Framework:** We introduce a UAV inspection framework that integrates global coverage planning with local adaptability to enable efficient and complete UAV inspection in partially unknown indoor environments.
- **Segment-based Global Coverage Planning:** We develop a segment-based viewpoint generation and sequencing method for efficient global coverage planning.
- **Inspection-oriented Local Adaptive Planning:** We develop a local planner that adapts both trajectory and view angle to maximize coverage under environmental uncertainty.

II. RELATED WORK

Autonomous UAV inspection methods can be broadly categorized by their reliance on prior environmental information. Mapless approaches do not require prebuilt maps and adapt online to unknown environments. Chen et al. [5] proposed a crack detection system using a Sliding Window Method (SWM) to generate trajectories from real-time images. An explore-then-exploit system in [6] conducts online 3D mapping before planning safe coverage paths. Similarly, in [7], exploration and inspection are conducted simultaneously using heterogeneous UAV. While these methods ensure up-to-date environmental awareness, they often suffer from longer execution times and lack global optimization, leading to redundant or inefficient coverage.

In contrast, map-based approaches assume access to a reference model and solve the coverage path planning (CPP) problem through two stages: viewpoint generation and path optimization. FC-Planner [1] samples viewpoints along a geometric skeleton to reduce redundancy, while ASSCPP [8]

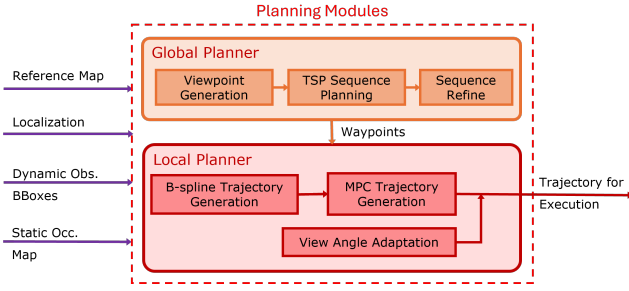


Fig. 2. System overview of the proposed adaptive inspection framework. The planning module takes the reference map, localization, and perception information as input and generates a collision-free trajectory as the output. The planning module includes a global planner for viewpoint generation and sequencing and a local planner for B-spline reference trajectory generation, MPC-based tracking and collision avoidance, and adaptive view angle adjustment to ensure coverage and safe navigation during execution.

adaptively refines sampling based on reconstruction quality. PredRecon [9] leverages surface prediction for faster and more targeted viewpoint selection. BIM-informed methods [10] incorporate robot configurations and spatial constraints to produce feasible plans in structured indoor settings. For path optimization, hierarchical methods [2] decompose global and local planning to improve scalability. Clustering-based frameworks [11] segment the inspection space for efficient TSP-based routing, while view resampling [12] enhances stability by smoothing redundant paths. Others integrate constraints like energy budgets using reinforcement learning [13], enabling CPP under realistic operational limits. However, most of these approaches assume static, fully known environments, limiting their robustness in dynamic or cluttered settings.

While prior work has addressed obstacle avoidance in navigation [14][15], few have incorporated obstacle awareness directly into the inspection objective. In contrast, our framework integrates global viewpoint planning with local obstacle-aware adaptation to improve coverage and efficiency in environments with unforeseen obstacles.

III. METHODOLOGY

Figure 2 shows an overview of the proposed framework. Provided by a reference map \mathcal{M}_R , the goal is to efficiently achieve the full coverage of \mathcal{M}_R using the onboard RGB-D camera for surface inspection. Localization is updated using LiDAR Inertial Odometry (LIO) [16], and perception for both static and dynamic obstacles is provided by the onboard perception module that takes the RGB-D image and LiDAR point cloud as inputs [17]. The proposed planning method consists of two parts: a global planner for inspection viewpoint generation (Sec. III-A) and a local planner for collision-free trajectory generation (Sec. III-B).

A. Segment-based Global Planner

The proposed global planner, as demonstrated in Alg. 1, takes a reference map \mathcal{M}_R as input and outputs a sequence of optimized viewpoints to maximize surface inspection coverage and efficiency.

The process begins with geometric segmentation of \mathcal{M}_R using the Region Growing method (Line 2), producing a set of segments \mathcal{S} . This segmentation clusters the environment based on surface normals and curvature, facilitating the subsequent generation of informative viewpoints and corresponding view angles. For each segment $s \in \mathcal{S}$, a set of candidate viewpoints \mathcal{V}_{raw} is generated to ensure comprehensive coverage (Lines 3-6). Since segmentation is conducted based on the surface normal, the viewpoints for each segment will be arranged in a linear spatial pattern. Each viewpoint v_i is defined as a tuple $v_i = (p_i, \phi_i)$. p_i denotes the position of the viewpoint, and ϕ_i is the view angle, which is initialized to be perpendicular to the principal orientation of the bounding box of the segment. The candidate viewpoints are then sequenced using the Lin-Kernighan-Helsgaun (LKH) solver [18] (Line 7). Then, a post-processing with a three-stage refinement procedure is conducted to handle the outliers and optimize the consistency. In the first stage of post-processing, a grouping is performed by mapping each viewpoint back to its source segment. Consecutive viewpoints from the same segment are merged into clusters, resulting in an initial cluster set \mathcal{C} (Line 8). In the second stage, clusters with sizes below a threshold τ_{max} are considered outliers and merged with the nearest adjacent cluster from the same source segment, provided that their combined sequence remains contiguous (Lines 9-14). This yields a refined set of clusters $\mathcal{C}_{\text{merged}}$. Finally, a local reordering is performed within each cluster to minimize intra-cluster travel distance while preserving the global cluster order to generate an optimized sequence of viewpoints $\mathcal{V}_{\text{optimized}}$, which serves as the basis for downstream local planning and inspection (Lines 15-16).

Algorithm 1: Global Planner

```

1  $\mathcal{M}_R \leftarrow$  Input Reference Map;
2  $\mathcal{S} \leftarrow \text{MapSegmentation}(\mathcal{M}_R) \triangleright$  segment
   bounding boxes
3  $\mathcal{V}_{\text{raw}} \leftarrow \emptyset \triangleright$  viewpoint array
4 for  $s \in \mathcal{S}$  do
5    $\mathcal{V}_s \leftarrow \text{GenerateViewpoints}(s) \triangleright$  viewpoint
   set for each segment
6    $\mathcal{V}_{\text{raw}}.\text{append}(\mathcal{V}_s)$ ;
7  $\mathcal{V}_{\text{ordered}} \leftarrow \text{SolveTSP}(\mathcal{V}_{\text{raw}}) \triangleright$  TSP viewpoint
   sequence
8  $\mathcal{C} \leftarrow \text{Remap}(\mathcal{V}_{\text{ordered}}, \mathcal{S}) \triangleright$  cluster array
9  $\tau \leftarrow$  Initial Threshold;
10  $\tau_{\text{max}} \leftarrow$  Predefined Maximum Threshold;
11 while  $\tau \leq \tau_{\text{max}}$  do
12    $\mathcal{C}_{\text{merged}} \leftarrow \text{MergeOutlier}(\mathcal{C}, \tau) \triangleright$  merged
   cluster array
13    $\mathcal{C} \leftarrow \mathcal{C}_{\text{merged}}$ 
14    $\tau \leftarrow \tau + 1$ 
15  $\mathcal{V}_{\text{optimized}} \leftarrow \text{LocalReorder}(\mathcal{C}_{\text{merged}}, \mathcal{S}) \triangleright$  output
   viewpoint sequence
16 return  $\mathcal{V}_{\text{optimized}}$ ;

```

B. Inspection-oriented Local Planner

The local planner integrates a static reference trajectory planner, a dynamic obstacle avoidance planner, and a local view angle planner to maximize inspection coverage and navigation safety.

Static Planner: Given a viewpoint as the navigation goal, a B-spline trajectory of order k is constructed from a set of control points:

$$\mathbb{P} = \{P_1, P_2, \dots, P_N\}, \quad P_i \in \mathbb{R}^3, \quad (1)$$

where the first and last $k - 1$ control points are fixed to the start and goal positions, respectively. The trajectory generation is formulated as an unconstrained optimization problem where the optimization variables are the set \mathbb{P} of the intermediate $N - 2(k - 1)$ control points, excluding the fixed start and goal. The objective is to minimize a cost function composed of weighted terms for control effort, trajectory smoothness, and static collision costs:

$$J(S) = \alpha_{\text{control}} \cdot J_{\text{control}} + \alpha_{\text{smooth}} \cdot J_{\text{smooth}} + \alpha_{\text{static}} \cdot J_{\text{static}}. \quad (2)$$

Details of the static planner can be found in our previous work [19]. The static planner generates a trajectory that avoids static obstacles, which serves as a reference for the downstream dynamic planner.

Dynamic Planner: Given a reference trajectory, we apply the model predictive control to generate final execution trajectories that avoid static and dynamic obstacles. The entire optimization problem can be formulated as:

$$\min_{\mathbf{x}_{0:N}, \mathbf{u}_{0:N-1}} \sum_{k=0}^N \|\mathbf{x}_k - \mathbf{x}_k^{\text{ref}}\|^2 + \sum_{k=0}^{N-1} \lambda_{\mathbf{u}} \|\mathbf{u}_k\|^2, \quad (3a)$$

$$\text{s.t.} \quad \mathbf{x}_0 = \mathbf{x}(t_0), \quad (3b)$$

$$\mathbf{x}_k = f(\mathbf{x}_{k-1}, \mathbf{u}_{k-1}), \quad (3c)$$

$$\mathbf{u}_{\min} \leq \mathbf{u}_k \leq \mathbf{u}_{\max}, \quad (3d)$$

$$\mathbf{x}_k \notin \mathcal{R}_i, \forall i \in \mathcal{O}_{\text{static}} \cup \mathcal{O}_{\text{dynamic}}, \quad (3e)$$

$$\forall k \in \{0, \dots, N\}, \quad (3f)$$

where $\mathbf{x}_k = [\mathbf{p}_k, \mathbf{v}_k]^T$ and $\mathbf{u}_k = \mathbf{a}_k$ represent the robot states and control inputs with the subscript indicating the time step. The objective (Eqn. 3a) is to minimize the deviation from the reference trajectory while using the least control effort. Equation 3b sets the initial state constraint based on the current robot states. The robot's dynamics model and control limits are presented by Eqns. 3c and 3d, respectively. The collision constraints (Eqn. 3e) ensure that the robot avoids collisions with static and dynamic obstacles. Details of the dynamic planner can be found in our previous work [20].

View-Angle Planner: During execution, viewpoints may become obstructed or unreachable due to unforeseen static obstacles in the environment, resulting in incomplete coverage. To mitigate this, the local planner incorporates an adaptive view angle strategy to maximize observable information in the presence of occlusions as demonstrated in Alg. 2.

The planner leverages the real-time occupancy map \mathcal{M} , incrementally updated from LiDAR point cloud data to

identify blocked viewpoints (Line 3). Given the camera field of view (FoV) and the nominal view angle of each obstructed viewpoint ϕ_i , the corresponding occluded regions on the reference map are inferred (Lines 4-6). Subsequently, a weight is assigned to each grid cell in the map \mathcal{M} (Line 7). While all grids are initialized with the same weight, lower weights are attributed to previously scanned regions, and higher weights are assigned to occluded or unobserved areas to reflect their observation status (Lines 8-12). Then, by utilizing the current robot state and the camera field of view, the planner performs discrete 360-degree raycasting on the map \mathcal{M} to evaluate the weighted visibility score across candidate view angles (Lines 13-15). The direction yielding the highest cumulative score is selected as the optimal view angle (Line 16). This adaptive mechanism enables the system to recover coverage lost to occlusions and enhances the overall robustness and efficiency of the inspection process.

Algorithm 2: Local View Angle Adaptation

```

1  $\mathcal{M} \leftarrow$  Input Occupancy Map;
2  $p_r \leftarrow$  Current Robot Position;
3  $\mathcal{V}_{\text{blocked}} \leftarrow \text{IdentifyBlockedViewpoints}(\mathcal{M})$ 
    $\triangleright$  obstructed viewpoints
4 for  $v_i \in \mathcal{V}_{\text{blocked}}$  do
5    $\phi_i \leftarrow$  nominal view angle of  $v_i$ ;
6    $\mathcal{G}_{\text{occluded}} \leftarrow \text{GetOccludedRegions}(v_i, \text{FoV}, \mathcal{M}_R)$ ;
7  $\mathcal{W} \leftarrow \text{InitializeGridWeights}(\mathcal{M}_R) \triangleright$  equal
   weight for all grids
8 for  $g \in \mathcal{M}_R$  do
9   if  $g \in \mathcal{G}_{\text{occluded}}$  then
10      $\mathcal{W}[g] \leftarrow \text{HigherWeight}()$ ;
11   else if  $g \in \mathcal{G}_{\text{scanned}}$  then
12      $\mathcal{W}[g] \leftarrow \text{LowerWeight}()$ ;
13  $\Phi \leftarrow$  Candidate View Angles;
14 for  $\phi \in \Phi$  do
15    $S[\phi] \leftarrow \text{RaycastScore}(\phi, p_r, \text{FoV}, \mathcal{W}, \mathcal{M})$ 
      $\triangleright$  weighted sum of visible grid
     scores
16  $\phi^* \leftarrow \arg \max_{\phi \in \Phi} S[\phi] \triangleright$  optimal view angle
17 return  $\phi^*$ ;
```

IV. RESULTS AND DISCUSSIONS

A. Simulation Experiments

To validate the proposed method, simulation experiments were conducted in two distinct indoor environments, as shown in Fig. 3. Each environment includes the inspection target and the unforeseen obstacles, such as static blocks and dynamic human workers, as shown in 1a) and 2a). Provided by the floor plan, 1b) and 2b) show the reference floor plan and the planned global path. The planned paths successfully visit all target viewpoints exactly once, ensuring comprehensive coverage with minimal redundancy. Although minor collisions may occur when transitioning between distant segments, the local planner generates the trajectory in real time to maintain collision-free navigation.

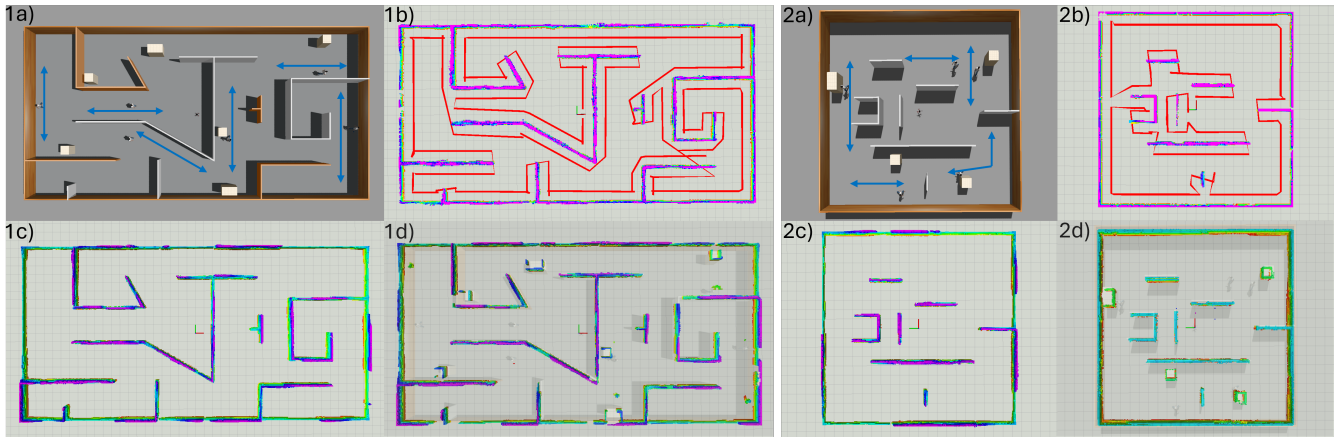


Fig. 3. Simulation results in two indoor environments for evaluating inspection coverage. 1a), 2a): Top-down views of the simulated environments with static and dynamic obstacles. Blue arrows show the movement directions of the dynamic obstacles. 1b), 2b): Planned global paths on the reference maps. 1c), 2c): Voxel grids show the coverage results of camera observations in static scenes (without unforeseen static and dynamic obstacles). 1d), 2d): Voxel grids show the coverage results in the environments shown in 1a) and 2a) and are visualized over the environment.

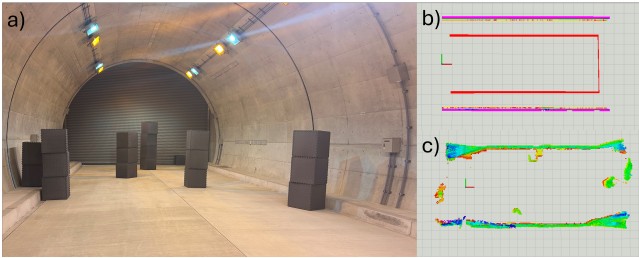


Fig. 4. Real-world inspection experiment in a tunnel environment. a) The physical test environment with multiple static obstacles. b) The reference map and the planned global paths. c) Voxel grids show the coverage results in the real flight.

Subfigures 1c) and 2c) demonstrate the inspection result in a static setting, in which the environment is the same as the provided reference map without obstacles involved. The grids indicate areas successfully captured by the onboard camera. These results show that following the planned trajectory can achieve full coverage of the inspection target. To further verify the inspection coverage under environmental uncertainty, static and dynamic obstacles (e.g., walking humans) were introduced into the scenes. As shown in 1d) and 2d), despite the discrepancy between the reference map and the real-time environment, our method successfully adapts to avoid unforeseen obstacles while preserving inspection quality, verifying its effectiveness in partially known and dynamic indoor environments.

B. Physical Flight Test

A physical flight test was conducted in a tunnel to validate the applicability of the proposed framework in real-world conditions. As shown in Fig. 4, the UAV performs an autonomous inspection task in the presence of multiple static obstacles. Subfigure a) presents the actual test environment, where the tunnel walls are to be inspected and blocks were placed to simulate unknown obstacles. The reference map, as shown in subfigure b), defines the inspection target without

informing the static obstacles. Despite this discrepancy, the coverage result shown in subfigure c) demonstrates that the UAV successfully maintains inspection coverage. The proposed method is able to avoid previously unknown obstacles while ensuring that the critical surfaces of interest are effectively captured, highlighting the robustness and real-world feasibility of the proposed planning framework.

The experimental results demonstrate that the proposed framework effectively achieves high inspection coverage and operational efficiency, even in environments with unforeseen static and dynamic obstacles. While the proposed method allows certain discrepancies between the reference map and the actual inspection environment and can avoid collision, full coverage cannot be guaranteed if the reference map deviates significantly from the true structure of the inspection target.

V. CONCLUSION AND FUTURE WORK

This work presents an adaptive UAV inspection framework that integrates global viewpoint generation and coverage planning with a hierarchical local planner. The global planner generates an efficient and complete inspection path based on a reference map, while the local planner ensures safe execution by incorporating both static and dynamic obstacle avoidance as well as real-time view angle adaptation. Simulation results in structured indoor environments demonstrate that our method is able to maintain high coverage rates under environmental uncertainty. Future work will focus on processing and analyzing the collected inspection data for application in infrastructure maintenance. Additionally, while the framework performs well in moderately sized environments, its scalability to large-scale industrial facilities remains to be further investigated.

REFERENCES

- [1] C. Feng, H. Li, M. Zhang, X. Chen, B. Zhou, and S. Shen, "Fc-planner: A skeleton-guided planning framework for fast aerial coverage of complex 3d scenes," in *2024 IEEE International Conference on Robotics and Automation (ICRA)*, 2024, pp. 8686–8692.

- [2] C. Cao, J. Zhang, M. Travers, and H. Choset, "Hierarchical coverage path planning in complex 3d environments," in *2020 IEEE International Conference on Robotics and Automation (ICRA)*, 2020, pp. 3206–3212.
- [3] Y. Wang, S. Wan, R. Qiu, and X. Li, "A path planning algorithm for uav 3d surface inspection based on normal vector filtering and integrated viewpoint evaluation," *Journal of Field Robotics*, 2024.
- [4] Z. Xu, B. Chen, X. Zhan, Y. Xiu, C. Suzuki, and K. Shimada, "A vision-based autonomous uav inspection framework for unknown tunnel construction sites with dynamic obstacles," *IEEE Robotics and Automation Letters*, vol. 8, no. 8, pp. 4983–4990, 2023.
- [5] X. Chen, C. Wang, C. Liu, X. Zhu, Y. Zhang, T. Luo, and J. Zhang, "Autonomous crack detection for mountainous roads using uav inspection system," *Sensors*, vol. 24, no. 14, p. 4751, 2024.
- [6] C. Gao, X. Wang, R. Wang, Z. Zhao, Y. Zhai, X. Chen, and B. M. Chen, "A uav-based explore-then-exploit system for autonomous indoor facility inspection and scene reconstruction," *Automation in Construction*, vol. 148, p. 104753, 2023.
- [7] X. Xu, M. Cao, S. Yuan, T. H. Nguyen, T.-M. Nguyen, and L. Xie, "A cost-effective cooperative exploration and inspection strategy for heterogeneous aerial system," in *2024 IEEE 18th International Conference on Control & Automation (ICCA)*. IEEE, 2024, pp. 673–678.
- [8] R. Almadhoun, T. Taha, D. Gan, J. Dias, Y. Zweiri, and L. Seneviratne, "Coverage path planning with adaptive viewpoint sampling to construct 3d models of complex structures for the purpose of inspection," in *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2018, pp. 7047–7054.
- [9] C. Feng, H. Li, F. Gao, B. Zhou, and S. Shen, "Predrecon: A prediction-boosted planning framework for fast and high-quality autonomous aerial reconstruction," in *2023 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2023, pp. 1207–1213.
- [10] Z. Chen, H. Wang, K. Chen, C. Song, X. Zhang, B. Wang, and J. C. Cheng, "Improved coverage path planning for indoor robots based on bim and robotic configurations," *Automation in Construction*, vol. 158, p. 105160, 2024.
- [11] J. Chen, C. Du, Y. Zhang, P. Han, and W. Wei, "A clustering-based coverage path planning method for autonomous heterogeneous uavs," *IEEE Transactions on Intelligent Transportation Systems*, vol. 23, no. 12, pp. 25 546–25 556, 2021.
- [12] H. Liu, Y. P. Tsang, C. K. Lee, and C. H. Wu, "Uav trajectory planning via viewpoint resampling for autonomous remote inspection of industrial facilities," *IEEE Transactions on Industrial Informatics*, 2024.
- [13] M. Theile, H. Bayerlein, R. Nai, D. Gesbert, and M. Caccamo, "Uav coverage path planning under varying power constraints using deep reinforcement learning," in *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2020, pp. 1444–1449.
- [14] X. Zhou, Z. Wang, H. Ye, C. Xu, and F. Gao, "Ego-planner: An esdf-free gradient-based local planner for quadrotors," *IEEE Robotics and Automation Letters*, vol. 6, no. 2, pp. 478–485, 2020.
- [15] J. Lin, H. Zhu, and J. Alonso-Mora, "Robust vision-based obstacle avoidance for micro aerial vehicles in dynamic environments," in *2020 IEEE International Conference on Robotics and Automation (ICRA)*, 2020, pp. 2682–2688.
- [16] W. Xu, Y. Cai, D. He, J. Lin, and F. Zhang, "Fast-lio2: Fast direct lidar-inertial odometry," *IEEE Transactions on Robotics*, vol. 38, no. 4, pp. 2053–2073, 2022.
- [17] Z. Xu, H. Shen, X. Han, H. Jin, K. Ye, and K. Shimada, "Lv-dot: Lidar-visual dynamic obstacle detection and tracking for autonomous robot navigation," *arXiv preprint arXiv:2502.20607*, 2025.
- [18] K. Helsgaun, "An effective implementation of the lin-kernighan traveling salesman heuristic," *European journal of operational research*, vol. 126, no. 1, pp. 106–130, 2000.
- [19] Z. Xu, Y. Xiu, X. Zhan, B. Chen, and K. Shimada, "Vision-aided uav navigation and dynamic obstacle avoidance using gradient-based b-spline trajectory optimization," in *2023 IEEE International Conference on Robotics and Automation (ICRA)*, 2023, pp. 1214–1220.
- [20] Z. Xu, H. Jin, X. Han, H. Shen, and K. Shimada, "Intent prediction-driven model predictive control for uav planning and navigation in dynamic environments," *IEEE Robotics and Automation Letters*, vol. 10, no. 5, pp. 4946–4953, 2025.