2026 IEEE International Conference on Robotics and Automation (ICRA 2026)
June 1-5, 2026. Vienna, Austria

Semantic and Terrain-Aware Trajectory Optimization for Uniform Coverage in Obstacle-Laden Environments

Zexuan Fan[1], Hengye Yang[1], Sunchun Zhou[1], Junyi Cai[1], Tao Sun[2,†], and Chang Liu[3,†]

***Abstract*— Achieving efficient and uniform coverage in obstacle-laden unknown environments is essential for autonomous robots in cleaning, inspection and agricultural operations. Unlike most existing approaches that prioritize path length and time optimality, we propose the SHIFT planner framework, which integrates semantic mapping, adaptive coverage planning, and real-time obstacle avoidance to ensure comprehensive coverage across diverse terrains and semantic features. We first develop an innovative Radiant-Field-Informed Coverage Planning (RFICP) algorithm, which generates trajectories that adapt to terrain variations. A Gaussian diffusion field is employed to adaptively adjust the robot's speed, ensuring efficient coverage under varying environmental conditions influenced by semantic attributes. Next, we present a novel incremental KD-tree sliding window optimization (IKD-SWOpt) method to effectively handle unforeseen obstacles. IKD-SWOpt leverages an enhanced A* algorithm guided by the IKD-tree distance field to generate initial local avoidance trajectories. Subsequently, it optimizes trajectory segments within and outside waypoint safety zones by evaluating and refining non-compliant segments using an adaptive sliding window. This method not only reduces computational overhead but also guarantees high-quality real-time obstacle avoidance. Extensive experiments were conducted using drones in simulated environments and robotic vacuum cleaners in real-world settings. The SHIFT planner demonstrates state-of-the-art performance in coverage uniformity and adaptability across various terrains while maintaining very low computational overhead.**

I. Introduction

The increasing reliance on autonomous robots across various fields has highlighted the need for advanced algorithmic frameworks that ensure both efficiency and comprehensive coverage of the task environment [1]–[3]. However, most existing methodologies prioritize minimizing traversal time and path length, often overlooking the critical need for uniform coverage and adaptability to dynamic conditions. Applications such as surface wiping [1], uniform pesticide spraying [2], and autonomous room cleaning [4] require optimized path coverage to achieve efficient and thorough results. Despite their widespread use, existing algorithms fail to fully address the task-specific coverage requirements—whether ensuring that the entire surface is cleaned thoroughly or maintaining accurate 3D reconstruction for inspections [5]. More comprehensive task models are essential for improving the real-world performance of these systems.

Coverage path planning (CPP) is a fundamental problem in robotics, aiming to determine a path for a robot to cover an entire space [6]. Classical methods, such as Boustrophedon Cellular Decomposition (BCD) [7] and Rapidly-Exploring Random Trees (RRT) [8], only ensure basic coverage feasibility and focus purely on geometric or kinematic aspects in 2D or pseudo-2D spaces. Over time, more sophisticated optimization-based CPP algorithms have emerged to balance coverage quality with runtime efficiency and to handle large, unstructured and dynamic environments with real-time constraints and challenging coverage objectives [5], [9]–[11].

However, most existing coverage algorithms rely on assumptions of simplified environments, often neglecting environmental attributes and 3D morphological factors. For instance, deformable spiral coverage algorithms in [11] generate smooth coverage paths through Deformable Spiral Coverage Path Planning (DSCPP), but they do not fully account for 3D morphological surfaces or cost-based cleaning validation. Similarly, the trochoid-based path planning algorithm for fixed-wing UAVs under wind disturbances proposed in [12] focuses solely on optimizing flight time, without considering terrain or mission-level objectives. Most existing offline coverage planning algorithms, such as Boustrophedon, grid-based TSP, and contour-line-based coverage, fail to incorporate real-time adjustments and semantic-level costs [3]. The primitive-based path planning approach, which utilizes path primitive sampling and coverage graphs to optimize distance and time, also does not fully address environmental attributes and 3D morphological factors [13]. In [9], a hierarchical decomposition approach is developed to divide the environment into manageable subregions. However, this method lacks explicit semantic or dynamic updates, resulting in feasible but suboptimal solutions in challenging circumstances.

Moreover, complex dynamic environments necessitate modern coverage strategies that can react to changes or obstacles in real-time through local re-planning and time allocation. In response to this requirement, [14] developed bio-inspired neural-network-based genetic algorithms capable of adapting coverage on-the-fly. However, these algorithms lack considerations for path continuity and time optimality. Similarly, [10] proposed a constriction decomposition method that focuses on time-optimized coverage and 3D reconstruction but does not explicitly model cleaning tasks as a function of coverage cost. Additionally, the open-source

[1] Robotics Perception and Planning Team, Fabric and Floor Care Division, Midea Group, China
[2] State Key Laboratory of High End Heavy Load Robots, China; Artificial Intelligence Research Center, Midea Group, China.
[3] School of Advanced Manufacturing and Robotics, Peking University, China
† Corresponding Authors
Emails:{zexuan.fan, yanghy228, zhousc18, caijy20, tsun}@midea.com, changliucoe@pku.edu.cn
Open-source implementation is available at https://fanzexuan.github.io/SHIFTPlanner/.

library Fields2Cover [15] provides a comprehensive CPP framework with ready-to-use coverage libraries supporting various types of curvature turns (e.g., Dubins and Reeds-Shepp). Nevertheless, it remains limited to 2D terrain and cannot address real-time obstacle avoidance or dynamic re-planning of smooth paths.

In conclusion, while many notable efforts have advanced coverage in specialized domains, the integration of 3D terrain, real-time re-planning, comprehensive semantic modeling (e.g., dirtiness as a cost function), and dynamic obstacle avoidance remains an active research challenge. Our work addresses these issues by proposing a unified framework, the SHIFT planner, which leverages a field-informed coverage approach with dynamic attribute-based speed control and an incremental sliding window optimization for real-time obstacle avoidance. Unlike the aforementioned methods that simplify 3D structures or treat complete coverage as a purely geometric challenge, our approach utilizes both local and global environmental attributes to regulate coverage speed and thoroughly consider obstacles and geometric constraints. Through the proposed RFICP and IKD-SWOpt approaches, we aim to ensure consistent coverage quality across non-uniform 3D terrain, maintain adaptability and path continuity in dynamic environments, and leverage semantic-level cues for cost-based planning. Extensive simulations and real-world experiments demonstrate the state-of-the-art performance of our SHIFT planner in terms of coverage uniformity and adaptability, while maintaining low computational overhead.

II. System Overview

The proposed SHIFT planner framework aims to achieve semantically aware coverage and real-time low-overhead obstacle avoidance in 3D environments. As illustrated in Fig. 1, SHIFT operates in two key stages: RFICP and IKD-SWOpt. We initially employ a differential geometry approach to fit a parametric surface to the sensor-derived point cloud. Subsequently, we compute and filter curvature metrics to eliminate irrelevant outliers and small-scale obstacles. The parameter domain of the smoothed surface is then discretized to create a uniform grid of points. Finally, the discretized point surface is integrated with local semantic information, such as dirtiness or dryness. As detailed in Section III, the RFICP module utilizes this fused data to generate a coverage trajectory that seamlessly adapts to 3D surface variations. Additionally, by leveraging a Gaussian diffusion field, RFICP adjusts the robot's speed based on semantic importance, allocating more time to highly soiled areas. Next, to handle dynamic obstacles, IKD-SWOpt is applied for local trajectory refinement, as illustrated in Section IV. The process begins with an A* search guided by an incremental KD-tree distance field, resulting in a promising local avoidance path. IKD-SWOpt then examines the trajectory segments within the safety distance of each waypoint. If a segment's safety score, which reflects collision risk and continuity, falls below a predetermined threshold, it is further optimized using an adaptive sliding window approach. This strategy significantly reduces computational overhead while ensuring rapid and collision-free path updates. Finally, the resulting trajectories are fed into a tracking controller to ensure accurate execution of the motion.

III. Radiant-Field-Informed Coverage Planning

To ensure efficient and uniform coverage, the initial environmental attribute level at location $\mathbf{p}$ is defined as:

$$C_i(\mathbf{p}) = kA(\mathbf{p}), \tag{1}$$

where k is a scaling coefficient, and $A(\mathbf{p})$ denotes the semantic field intensity. Our objective is to reduce this initial attribute to a target level C_t. Consequently, the required effective coverage C_e is derived as the difference between the initial level C_i and the target:

$$C_e(\mathbf{p}) \;=\; C_i(\mathbf{p}) - C_t \;=\; kA(\mathbf{p}) - C_t. \tag{2}$$

Since coverage operations are only necessary where the initial attribute exceeds the target, we focus on the domain where $kA(\mathbf{p}) > C_t$ in this paper.

Then, we model the spatial influence of the robot at position $\mathbf{p}'$ on the surrounding environment at position $\mathbf{p}$ using a 2D Gaussian diffusion kernel:

$$G(\mathbf{p};\mathbf{p}') \;=\; \frac{1}{2\pi\sigma^2} e^{-\frac{\|\mathbf{p}-\mathbf{p}'\|^2}{2\sigma^2}}, \tag{3}$$

where $\|\mathbf{p}-\mathbf{p}'\| \le R$. For computational efficiency, we we restrict the influence range to a circle of radius R. Consequently, when the robot dwells at position $\mathbf{p}'$ for a duration t, its accumulated coverage at position $\mathbf{p}$ can be modeled as an integration of the Gaussian kernel within this neighborhood:

$$C_{\text{int}}(\mathbf{p}) = \int_{\|\mathbf{p}-\mathbf{p}'\|\le R} G(\mathbf{p};\mathbf{p}') \left[1 - e^{-\lambda\, t(\mathbf{p}')}\right] d\mathbf{p}', \tag{4}$$

where λ is a coverage efficiency constant. We require the accumulated coverage $C_{\text{int}}(\mathbf{p})$ to match the required level $C_e(\mathbf{p})$ to effectively reduce the environmental attribute to the target.

Assuming an approximately uniform dwell time $t(\mathbf{p})$ within the local Gaussian footprint, and by combining (2) and (4), we obtain:

$$\begin{aligned} kA(\mathbf{p}) - C_t &= \left[\int_{\|\mathbf{p}-\mathbf{p}'\|\le R} G(\mathbf{p};\mathbf{p}')\, d\mathbf{p}'\right] \left[1 - e^{-\lambda\, t(\mathbf{p})}\right] \\ &= \left(1 - e^{-\frac{R^2}{2\sigma^2}}\right) \left[1 - e^{-\lambda\, t(\mathbf{p})}\right], \end{aligned} \tag{5}$$

where the first bracketed term represents the integration of the normalized 2D Gaussian kernel over a disk of radius R. The dwell time $t(\mathbf{p})$ can then be obtained as

$$t(\mathbf{p}) = -\frac{1}{\lambda} \ln\left[1 - \frac{kA(\mathbf{p}) - C_t}{1 - e^{-\frac{R^2}{2\sigma^2}}}\right]. \tag{6}$$

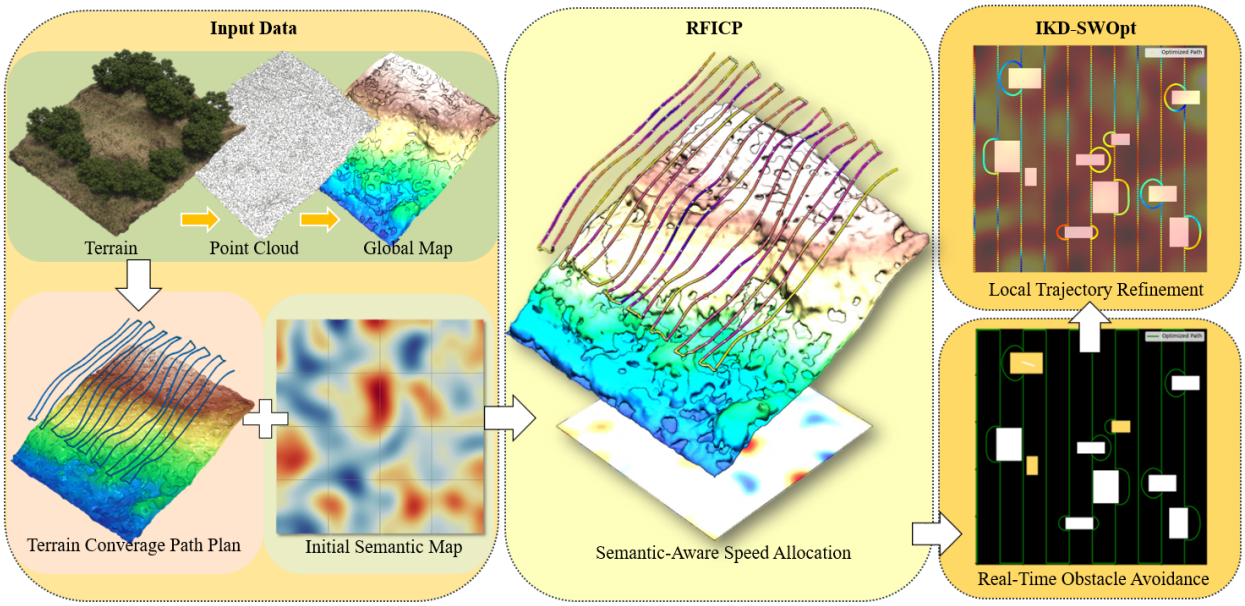


Fig. 1: Overall architecture of the SHIFT planner. The terrain-derived point cloud data is initially filtered and integrated into a global map. This map, combined with a semantic map, serves as input for the global planner. RFICP then generates an adaptive coverage trajectory by adjusting the robot's speed based on semantic information, while IKD-SWOpt performs real-time refinements of local trajectories when obstacles are encountered.

Since the traversal speed v is inversely proportional to the dwell time, it can be computed as:

$$v(\mathbf{p}) = \frac{\Delta s}{t(\mathbf{p})} = -\frac{\lambda \Delta s}{\ln\left[1 - \frac{k\,A(\mathbf{p}) - C_t}{1 - e^{-R^2/2\sigma^2}}\right]}. \tag{7}$$

where Δs represents the length of the discretized path segment. In practice, we clamp $v(\mathbf{p})$ to lie within the robot's physical velocity limits (e.g., to handle singularities when coverage demand is negligible) and enforce acceleration constraints using standard velocity interpolation techniques.

IV. Real-Time Obstacle Avoidance

While RFICP ensures comprehensive coverage and speed allocation, unforeseen obstacles can invalidate parts of the trajectory. Within our SHIFT planner framework, we develop the IKD-SWOpt method to refine local segments upon detecting collision or near-collision states, thereby achieving real-time obstacle avoidance.

A. A* Search with IKD-Tree-Informed Distance Field

In partially unknown environments, a continuously updated incremental k-dimensional tree (IKD-tree) is maintained to represent obstacle locations [16]. We build a distance field $\mathcal{D}(\mathbf{p})$ from the IKD-tree, which represents the distance from the waypoint $\mathbf{p}$ to the nearest obstacle. Given the distance field $\mathcal{D}$, the initial A* search uses the cost function below:

$$f(\mathbf{p}) \;=\; g(\mathbf{p}) + h(\mathbf{p}) + \alpha \cdot \frac{1}{\mathcal{D}(\mathbf{p}) + \varepsilon}, \tag{8}$$

where $g(\mathbf{p})$ and $h(\mathbf{p})$ are the usual path cost and heuristic terms respectively, α is a weighing factor, and ε is a small constant for numerical stability. In our implementation, the parameters are empirically set to $\alpha = 0.5$ and $\varepsilon = 0.01$ to ensure proper cost scaling. The third term in (8) penalizes positions close to obstacles. The IKD-tree provides near-lossless distance-field lookups, enhancing A* heuristics and avoiding local minima. Therefore, this enhanced IKD-tree-informed A* algorithm generates a nice initial avoidance path with better clearance. We then use this path as a starting point for subsequent local gradient-based optimization.

B. Non-Compliant Segment Identification

Let $\mathcal{P} = \{\mathbf{p}_k\}_{k=1}^{N}$ be the initial trajectory obtained by the enhanced A* search. For each waypoint $\mathbf{p}_k$, we define a circular safety region

$$\mathcal{C}_k \;=\; \Big\{\mathbf{p} \;\Big|\; \|\mathbf{p} - \mathbf{p}_k\| \le r_{\text{safe}}\Big\}, \tag{9}$$

where the radius r_{safe} can be obtained as the distance to the nearest obstacle in the IKD-tree.

Given all nearby waypoints in the circular safety region, we define the safety score for $\mathbf{p}_k$ as

$$S_k = \sum_i f[\mathcal{D}(\mathbf{p}_i), C_{\text{cont}}(\mathbf{p}_i), C_{\text{feas}}(\mathbf{p}_i)], \text{ for } \mathbf{p}_i \in \mathcal{C}_k, \tag{10}$$

where $f(\cdot)$ is a weighted combination function, $\mathcal{D}$ represents the distance to the nearest obstacle, C_{cont} denotes the difference in headings or curvatures between consecutive waypoints, and C_{feas} is a feasibility metric indicating collisions or violations of dynamic constraints, such as velocity or acceleration limits. If S_k falls *below* a specified safety threshold τ_{safe}, the local path segment is considered *non-compliant*. We then mark the surrounding trajectory segment

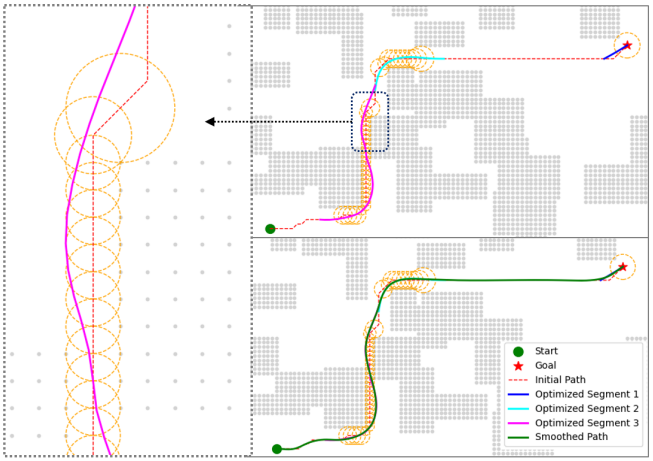


Fig. 2: An illustration of the local segment identification and refinement. For each waypoint on the initial path (red), a circular region (yellow) is defined. We then gather all identified non-compliant segments whose safety score falls below the threshold into an adaptive sliding window for local optimization.

within the circular safety region, denoted by $\mathcal{P}_{k-\Delta:k+\Delta}$, for local re-optimization. As shown in Fig.2, this method ensures that both the immediate vicinity of $\mathbf{p}_k$ and its neighboring waypoints are jointly refined. All identified non-compliant segments are gathered into an *adaptive sliding window* for subsequent optimization. The outline of the non-compliant segment identification is presented in Algorithm 1.

Algorithm 1 Non-Compliant Segment Identification

$\mathcal{S}_{\text{nc}} \leftarrow \emptyset$
for each $\mathbf{p}_k \in \mathcal{P}$ **do**
  Define $\mathcal{C}_k = \{\mathbf{p} \mid \|\mathbf{p} - \mathbf{p}_k\| \leq r_{\text{safe}}\}$
  Extract waypoints nearby $\{\mathbf{p}_i \in \mathcal{C}_k\}$
  Compute safety score S_k via (10)
  if $S_k < \tau_{\text{safe}}$ **then**
    $\mathcal{P}_{k-\Delta:k+\Delta} \leftarrow \{\mathbf{p}_i \in \mathcal{C}_k\}$
    $\mathcal{S}_{\text{nc}} \leftarrow \mathcal{S}_{\text{nc}} \cup \{\mathcal{P}_{k-\Delta:k+\Delta}\}$
  end if
end for
return $\mathcal{S}_{\text{nc}}$

C. IKD-Tree Sliding Window Optimization

Given that each segment already has a well-informed initial path obtained from the A* search, our solver refines only the control points within the window while maintaining global continuity. The local objective function is formulated as:

$$J = \sum_{i=1}^{N} \left[\frac{\beta_1}{\mathcal{D}(\mathbf{p}_i) + \varepsilon} + \beta_2 \|\mathbf{p}_{i+1} - 2\mathbf{p}_i + \mathbf{p}_{i-1}\|^2 + \beta_3 \|\mathbf{p}_{i+1} - \mathbf{p}_i\| \right], \tag{11}$$

where the three terms represent the obstacle clearance penalty, smoothness regularization, and path length cost, respectively. The weights β_1, β_2, and β_3 balance these objectives for the N points within the current segment S_{nc}. We then employ a gradient-based solver, such as the limited-memory BFGS (L-BFGS) method [17], to minimize (11) for the waypoints within the sliding window. This effectively pushes the path away from obstacles while maintaining coverage constraints. Additionally, this segmented optimization is highly parallelizable and can leverage GPU acceleration. By distributing the optimization of segments across GPU cores, we can significantly reduce computation time and improve real-time obstacle avoidance in dynamic environments.

Each time a window is optimized, the updated segment is merged back into the global trajectory. We then apply B-spline fitting to smooth the transition between original and optimized segments, and the smoothed trajectory is given by

$$\mathbf{C}(t) = \sum_{i=0}^{n} \mathbf{p}_i \, N_{i,k}(t), \tag{12}$$

where $\mathbf{p}_i$ represents the combined control points from both segments, and $N_{i,k}$ is the base function. This continuous spline ensures feasible velocities and accelerations at junctions and avoids abrupt turns. The general flow of the IKD-SWOpt algorithm is presented in Algorithm 2. It efficiently handles dynamic obstacles by incrementally updating the KD-tree with new obstacle information and refining only the affected segments of the trajectory.

Algorithm 2 IKD-SWOpt

while new obstacle data arrives **do**
  Update IKD-tree: Insert new obstacles into $\mathcal{T}$
  Segment Identification: Obtain $\mathcal{S}_{\text{nc}}$ from Algorithm 1
  for each segment $\mathcal{S} \in \mathcal{S}_{\text{nc}}$ **do**
    Sliding Window Optimization: Optimize $\mathcal{S}$ by minimizing J in (11)
  end for
  Trajectory Reconnection: Merge optimized segments into initial path $\mathcal{P}_0$ using B-spline smoothing
end while
return $\mathcal{P}_{\text{opt}}$

V. SHIFT Planner Implementation

The SHIFT planner integrates terrain-aware coverage planning (RFICP) with an incremental obstacle avoidance framework (IKD-SWOpt). Initially, a parametric surface $\mathbf{S}(u, v)$ is reconstructed from raw LiDAR data, utilizing curvature metrics for outlier rejection. This smoothed surface is discretized into a grid of waypoints $\{\mathbf{p}_{i,j}\}$, which are offset to maintain a uniform coverage height and sequenced using a boustrophedon pattern. Within the RFICP module, the semantic field intensity $A(\mathbf{p})$ is evaluated for each waypoint to generate an initial coverage trajectory $\mathcal{P}_0$, incorporating semantic-driven velocity profiles $v(\mathbf{p})$. To handle environmental changes, an IKD-tree $\mathcal{T}$, initialized with known obstacles, incrementally updates the distance field $\mathcal{D}(\mathbf{p})$. Inside the IKD-SWOpt module, paths are initialized via a local A* search guided by IKD-tree heuristics. Segments violating safety constraints, denoted as $\mathcal{S}_{\text{nc}}$, are identified through circular safety region

checks and subsequently refined within an adaptive sliding window using the BFGS algorithm. This ensures obstacle clearance, path smoothness, and efficiency. Finally, B-spline interpolation reconnects the segments to yield the optimized trajectory $\mathcal{P}_{\text{opt}}$, ensuring global continuity and uniform coverage.

VI. Experiments and Results

The proposed SHIFT planner has been validated through both numerical simulations (Section VI-A) and physical experiments (Section VI-B). In each section, we first describe our experimental setup and implementation details. Subsequently, we compare the coverage performance of our planning algorithms across various semantic attributes and evaluate the real-time capabilities of local trajectory optimization in the presence of obstacles against several benchmark state-of-the-art planning algorithms.

A. Numerical Simulations

We simulate an agricultural drone performing coverage operations in outdoor terrains measuring 10 meters by 15 meters, with non-uniform aridity levels as depicted in the semantic map in Fig. 3. During drone irrigation, the spacing between Boustrophedon paths is manually set to 1 meter, and the drone's altitude is consistently maintained at 1.5 meters above the ground. In the simulation environment, trees are represented by dark green obstacles, with their number, radius, and height randomly generated. The simulation is conducted on a computer equipped with an Intel i7 CPU running at 3.4 GHz and 16 GB of RAM. The drone's onboard LiDAR and camera systems provide terrain point clouds and semantic information through sensor fusion and segmentation. Point cloud processing, including surface extraction and curvature filtering, is implemented in C++ to ensure real-time efficiency. The IKD-tree is dynamically updated as new obstacle data becomes available, enabling near-lossless distance queries for local re-planning. Fig. 3 illustrates how the IKD-SWOpt method locally refines the drone's path to bypass obstacles and adjusts its speed in real time based on local aridity levels, thereby ensuring the uniform irrigation of drier crops.

Furthermore, we compare the IKD-SWOpt local trajectory refinement capability with two well-known methods based on Euclidean Signed Distance Field (ESDF), the Fast-Planner [18] and EWOK [19]. Fast-Planner is an efficient kinodynamic planner with a front-end graph search and a back-end gradient-based optimization over an ESDF. EWOK is a B-spline–based local planner that updates a 3D occupancy buffer and ESDF in real time. The comparison is conducted using the same three-dimensional terrain map as described above, incorporating randomly placed cylindrical obstacles of varying heights. With the obstacle density ranging from 0.1 to 0.5 obs/m^2, the local re-planning capabilities of these three planners are compared in terms of total flight time t, path length L, energy consumption E, and per-iteration planning time t_{p}. Notably, the energy consumption is normalized against the maximum observed value (i.e., that of the EWOK method). Additionally, the performance of IKD-SWOpt with and without GPU acceleration are both evaluated to demonstrate its computational efficiency and scalability. The GPU-accelerated experiments are conducted using an NVIDIA Quadro P620. Table I presents the average performance metrics among several runs at a medium obstacle density of 0.3 obs/m^2.

TABLE I: Comparison of Local Planning Abilities

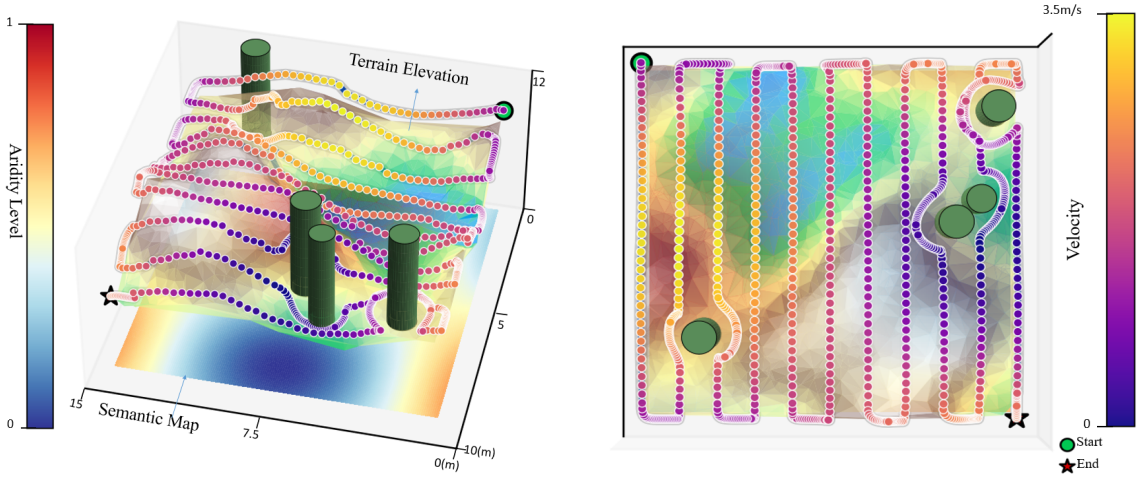
Method	t(s)	L(m)	E (Norm.)	t_{p}(ms)	**GPU Accel.**
EWOK	22.3	38.7	1.00	3.1	Off
Fast-Planner	21.6	35.4	0.75	4.2	Off
IKD-SWOpt	21.9	36.2	0.81	1.6	Off
IKD-SWOpt	21.5	35.9	0.79	0.9	On

IKD-SWOpt without GPU acceleration requires no explicit ESDF construction, reducing the per-iteration planning time t_{p} to 1.6 ms, which is less than half of the EWOK and Fast-Planner's planning time on average. With GPU acceleration, t_{p} is significantly further reduced to 0.9 ms. The path length L and total flight time t of IKD-SWOpt are comparable to Fast-Planner's kinodynamic search, indicating efficient path planning without the overhead of repeated ESDF updates. Although IKD-SWOpt without GPU acceleration has 8% higher energy consumption compared to Fast-Planner, the GPU-accelerated version reduces energy cost to 150.3 normalized units, keeping it competitive. The multi-segment sliding window optimization approach in IKD-SWOpt is highly parallelizable, allowing simultaneous optimization of multiple trajectory segments on GPU cores. This parallelization significantly reduces computation time, enables real-time trajectory adjustments and enhances the planner's responsiveness in dynamic environments.

B. Physical Experiments

As illustrated in Fig. 4, we validated our SHIFT planning method using the Eureka J15pro robotic vacuum cleaner. The robot is equipped with a LiDAR, front-facing camera, and IMU for mapping and localization, and operates within a dirtiness-labeled indoor environment deployed with cylindrical obstacles. Semantic information is obtained in real time through PanoOcc, a camera-based 3D panoptic segmentation approach that leverages spatiotemporal data from multi-frame and multi-view images captured by the front-facing camera [20]. We integrated a motion tracking system to automatically compute the coverage ratio in real-time. The SHIFT Planner intelligently allocates extended dwell times to dirtier zones, while the IKD-SWOpt module promptly responds to unforeseen obstacles. Multiple runs within the same indoor environment demonstrate that the SHIFT Planner achieves comprehensive semantic coverage and enables low-latency obstacle avoidance, making it highly suitable for complex real-world scenarios. Additional details regarding the physical experiments are provided in the accompanying video.

The coverage performance of the proposed SHIFT planner is evaluated against two sets of baselines: ablation variants



(a) Side View (b) Top-Down View

Fig. 3: Drone irrigation simulation results: the SHIFT planner generates coverage trajectories that adapt the drone's velocity to varying terrains and aridity levels while locally refining the path to avoid unforeseen obstacles (shown in green).

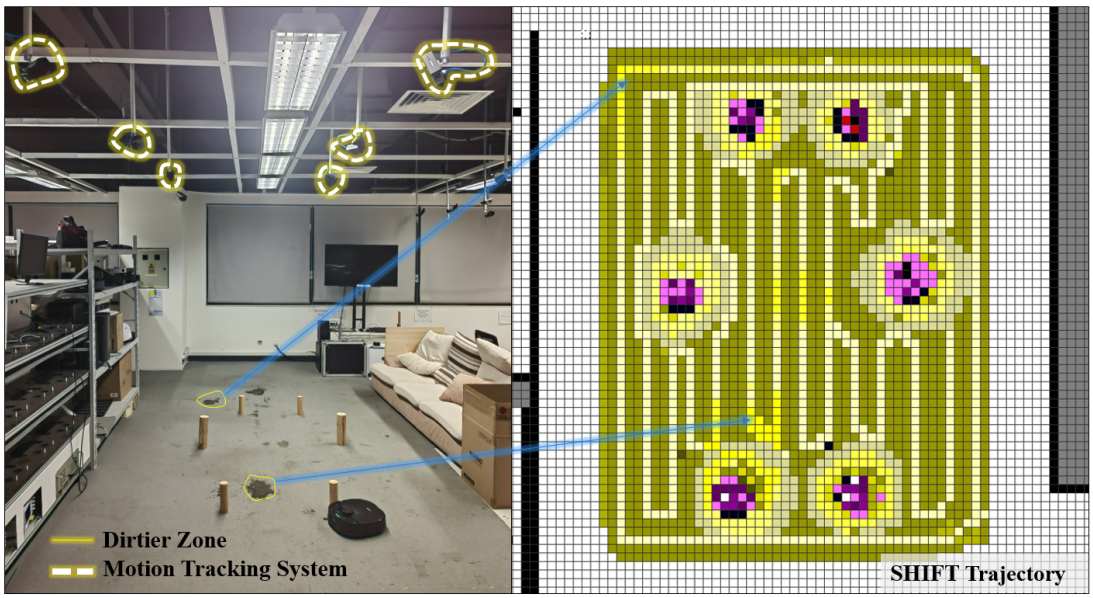


Fig. 4: A vacuum robot navigates a semantically labeled indoor environment, decelerating in dirtier zones (yellow) and promptly adapting to avoid unforeseen cylindrical obstacles.

and state-of-the-art (SOTA) approaches. For the ablation study, we test versions without RFICP (implying a uniform speed profile) and without IKD-SWOpt (omitting trajectory refinement for obstacle avoidance). Additionally, we benchmark our method against two SOTA approaches: CCPP [21] and K-Means+DRL [22]. Table II presents a comparative analysis based on four key performance metrics, averaged over 10 independent runs. The *coverage completeness* (C) is defined as the ratio of the cleaned area to the total accessible area. The *path overlap rate* (R_{ov}) is represented as the proportion of the path re-traversing cleaned areas. The *energy efficiency* (ϵ) is defined as the cleaned area per unit of energy consumed, normalized against the maximum value observed (i.e., that of the SHIFT method), such that 1 represents optimal efficiency. The *weighted cleaning uniformity* (U_w) evaluates whether the cleaning effort matches the dirtiness distribution. Specifically, we define a satisfaction ratio $r_i = \rho c_i / A_i$, where ρ is a scaling factor, c_i is the actual visit count, and A_i is the semantic intensity for grid i. U_w is then calculated based on the coefficient of variation (CV) of $\{r_i\}$:

$$U_w = 1 - \frac{\sigma_r}{\mu_r}, \tag{13}$$

where μ_r and σ_r are the mean and standard deviation of $\{r_i\}_{i=1}^{N}$. A value of U_w closer to 1 indicates that the cleaning intensity is distributed more proportionally to the local environmental demand.

TABLE II: Coverage Performance Comparison

Performance Metric	SHIFT Planner			CCPP	K-Means+DRL
	RFICP+IKD-SWOpt	IKD-SWOpt	RFICP		
Coverage Completeness C (%)	**98.6**	96.3	97.0	96.8	94.5
Path Overlap Rate R_{ov} (%)	**4.3**	4.6	7.8	8.7	12.1
Energy Efficiency ϵ (Norm.)	**1.00**	0.91	0.95	0.90	0.84
Weighted Cleaning Uniformity U_w (%)	**98.2**	90.2	97.6	90.1	86.2

Leveraging semantic-driven speed modulation (RFICP) and local trajectory optimization (IKD-SWOpt), the SHIFT planner achieves a 51% reduction in path overlap compared to the CCPP method and an 13% improvement in weighted cleaning uniformity over K-Means+DRL, while maintaining superior coverage completeness and energy efficiency. In contrast, the competing SOTA approaches fail to adequately clean heavily soiled regions, despite covering the entire space. The ablation study further reveals that RFICP significantly improves weighted cleaning uniformity by prolonging dwell time via deceleration in heavily soiled areas, while IKD-SWOpt markedly reduces path overlap rates by ensuring effective real-time obstacle avoidance. Consequently, the SHIFT planner not only guarantees higher-quality coverage but also minimizes path redundancy, yielding superior overall efficiency.

VII. Conclusion

In this paper, we introduce the SHIFT planner, a novel coverage planning framework designed for autonomous robots in complex and dynamic environments. It combines semantic mapping, terrain-aware coverage planning, and dynamic obstacle avoidance to achieve efficient and uniform coverage. Our RFICP algorithm adjusts speed and trajectory using a diffusion field model, allowing adaptive and consistent coverage across various environmental conditions. Additionally, the IKD-SWOpt efficiently handles obstacle avoidance by locally refining trajectories within a sliding detection window, ensuring safe and responsive navigation. Extensive experiments demonstrate the efficiency and adaptability of our framework in achieving high-quality coverage under real-world constraints. Comparative analyses indicate that the SHIFT planner outperforms state-of-the-art methods in terms of coverage completeness, path efficiency, and energy consumption. In the future, we plan to extend this approach to multi-robot systems, enabling collaborative coverage in larger and more dynamic environments.

References

[1] R. K. Megalingam, S. R. R. Vadivel, S. S. Kotaprolu, B. Nithul, D. V. Kumar, and G. Rudravaram, "Cleaning robots: A review of sensor technologies and intelligent control strategies for cleaning," *Journal of Field Robotics*, 2025.

[2] A. Hafeez, M. A. Husain, S. Singh, A. Chauhan, M. T. Khan, N. Kumar, A. Chauhan, and S. Soni, "Implementation of drone technology for farm monitoring & pesticide spraying: A review," *Information Processing in Agriculture*, vol. 10, no. 2, pp. 192–203, 2023.

[3] G. Mier, J. ao Valente, and S. de Bruin, "Indoor coverage path planning: Survey, implementation, analysis," *Journal of Field Robotics*, vol. 40, no. 5, pp. 765–789, 2023.

[4] M.-C. Kang, K.-S. Kim, D.-K. Noh, J.-W. Han, and S.-J. Ko, "A robust obstacle detection method for robotic vacuum cleaners," *IEEE Transactions on Consumer Electronics*, vol. 60, no. 4, pp. 587–595, 2014.

[5] 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)*, pp. 8686–8692, IEEE, 2024.

[6] E. Galceran and M. Carreras, "A survey on coverage path planning for robotics," *Robotics and Autonomous Systems*, vol. 61, no. 12, pp. 1258–1276, 2013.

[7] H. Choset, "Coverage of known spaces: The boustrophedon cellular decomposition," *Autonomous Robots*, vol. 9, no. 3, pp. 247–253, 2000.

[8] S. LaValle, "Rapidly-exploring random trees: A new tool for path planning," *Research Report*, 1998.

[9] 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)*, pp. 3206–3212, IEEE, 2020.

[10] S. Brown and S. L. Waslander, "The constriction decomposition method for coverage path planning," *IEEE Robotics and Automation Letters*, vol. 8, no. 4, pp. 3233–3238, 2023.

[11] M. Turner and S. Robinson, "A deformable spiral-based algorithm to smooth coverage path planning for marine growth removal," *Ocean Engineering*, vol. 250, p. 112045, 2023.

[12] A. Johnson and R. Lee, "Flight testing boustrophedon coverage path planning for fixed wing uavs in wind," *Aerospace Science and Technology*, vol. 158, pp. 1124–1137, 2023.

[13] R. Clark and L. Wright, "Coverage path planning using path primitive sampling and primitive coverage graph for visual inspection," *Automation in Construction*, vol. 155, p. 104052, 2023.

[14] J. Doe and J. Smith, "Online complete coverage path planning of a reconfigurable robot using glasius bio-inspired neural network and genetic algorithm," *Robotics and Autonomous Systems*, vol. 155, pp. 1043–1058, 2023.

[15] G. Mier, J. ao Valente, and S. de Bruin, "Fields2cover: An open-source coverage path planning library for unmanned agricultural vehicles," *IEEE Robotics and Automation Letters*, vol. 8, no. 4, pp. 2167–2172, 2023.

[16] W. Xu, Y. Cai, D. He, J. Lin, and F. Zhang, "Fast-lio2: Fast direct lidar-inertial odometry," *IEEE Transactions on Robotics*, 2022.

[17] 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.

[18] B. Zhou, F. Gao, L. Wang, C. Liu, and S. Shen, "Fast-planner: An efficient coverage path planning framework for unmanned aerial vehicles," *IEEE Robotics and Automation Letters*, vol. 8, no. 4, pp. 3529–3536, 2019.

[19] V. Usenko, L. Von Stumberg, A. Pangercic, and D. Cremers, "Real-time trajectory replanning for mavs using uniform b-splines and a 3d circular buffer," in *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 215–222, IEEE, 2017.

[20] Y. Wang, Y. Chen, X. Liao, L. Fan, and Z. Zhang, "Panoocc: Unified occupancy representation for camera-based 3d panoptic segmentation," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 17158–17168, 2024.

[21] 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. 25546–25556, 2021.

[22] J. Ni, Y. Gu, G. Tang, C. Ke, and Y. Gu, "Cooperative coverage path planning for multi-mobile robots based on improved k-means clustering and deep reinforcement learning," *Electronics*, vol. 13, no. 5, p. 944, 2024.