Koc. J. Sci. Eng., 8(2): (2025) 154-162 https://doi.org/10.34088/kojose.1694778

Kocaeli University

Kocaeli Journal of Science and Engineering

http://dergipark.org.tr/kojose

Coverage and Transition Planning in Obstacle-Rich Workspaces

Suat KARAKAYA [1, *]

[1] *Department of Mechatronics Engineering, Kocaeli University, Kocaeli, Türkiye,* ***ORCID:*** *0000-0002-3082-0304*

Article Info

Research paper

Received : *May 7, 2025*
Accepted : *July 10, 2025*

Keywords

Coverage Planning
Mobile Robots
Cell Partitioning
Path Optimization
A Algorithm*
Obstacle Avoidance

Abstract

This study proposes a comprehensive and efficient algorithm for coverage path planning, specifically designed for autonomous mobile robots operating on two-dimensional grid-based maps with fixed obstacles. The suggested approach merges several established techniques into a cohesive framework, encompassing row-based cellular decomposition, A*-based safe transition planning, and the dynamic optimization of entry and exit points for each segment of coverage. This synthesis allows for the creation of thorough and secure coverage paths while minimizing redundancy. The algorithm undergoes evaluation on maps of varying intricacy (5×5, 15×15, and 50×50), and its performance is measured using various metrics, including coverage density, average cell length, A* transition rate, and overall path efficiency. Results indicate that the proposed integration-based method consistently achieves high levels of coverage and path efficiency, while ensuring safe transitions between different areas. This framework proves to be not only practically efficient but also flexible across different scales, showcasing significant potential for real-world uses such as autonomous cleaning, agricultural monitoring, and emergency rescue operations.

1. Introduction

Autonomous mobile robots have attracted considerable research attention due to their ability to execute tasks independently across various fields, such as cleaning, agriculture, exploration, and search-and-rescue operations. Within these fields, Coverage Path Planning (CPP) has emerged as a significant challenge that necessitates robots to methodically navigate and completely scan a designated environment. The main goals of CPP are to guarantee total coverage while minimizing either the overall distance traveled or the time required for operations [1,2]. In static, enclosed settings, CPP usually assumes prior awareness of the workspace, including the locations of fixed obstacles [3,4]. Given this assumption, robots must attain both comprehensive coverage and movement that avoids collisions.

To tackle this issue, numerous techniques for environment decomposition have been suggested, dividing the workspace into separate cells based on free and occupied regions. Among these methods, the Boustrophedon decomposition technique is notable for its efficiency in segmenting free areas, allowing for systematic navigation by robots [1,5]. An essential component of CPP is identifying the optimal order of cell visits, an aspect frequently overlooked in earlier research but later acknowledged for its effect on coverage efficiency [6,7]. To enhance cell ordering, researchers have restructured the problem as a variant of the Traveling Salesman Problem (TSP), leading to various solution strategies [8–10].

Additionally, the entry and exit points for each cell also affect the overall path length, making meticulous optimization essential [11,5]. Conventional methods often set entry/exit directions rigidly, but recent advancements show that adjusting these parameters dynamically can further decrease coverage time [11,5]. For example, hierarchical TSP-based approaches, like those proposed by Xu et al. [8], optimize both intra-cell and inter-cell transitions for better performance. However, direct linear transitions between cells may introduce collision risks in areas dense with obstacles. To alleviate this, the A* algorithm is commonly used to determine the shortest obstacle-free paths between waypoints, ensuring both efficiency and safety [12,13].

* Corresponding Author: suat.karakaya@kocaeli.edu.tr

The exploration of multi-robot systems has also been conducted to improve coverage efficiency via collaborative task assignment and coordination [14–17]. Importantly, persistent monitoring techniques, such as those developed by Smith et al. [18], allow for ongoing coverage in dynamic environments over long durations.

Literature Review:

Grid-based CPP has been further advanced with methods like exact and approximate cellular decomposition [1,5], spanning-tree coverage techniques [6], and heuristic enhancements such as zig-zag and spiral patterns [7]. Graph-based search algorithms like A* [12] and D* [13] have been crucial in ensuring collision-free transitions between cells. Recent studies have examined multi-robot coordination [14–17], adaptive online replanning [19], UAV-specific CPP solutions [20], energy-efficient path planning [21], and 3D space coverage [22]. However, few investigations holistically integrate local cell coverage, entry/exit optimization, and inter-cell transitions in a single framework, especially in environments with dense obstacles.

Novelty:

The primary contribution of this study is the creation of a comprehensive and cohesive coverage path planning framework that effectively combines cell decomposition, dynamic entry-exit point optimization, and A*-based safe transitions for grid environments with high obstacle density. While prior research has generally addressed these components separately, our methodology uniquely integrates local and global planning elements into a unified system, overcoming significant limitations found in traditional approaches. More specifically, the proposed algorithm enhances the current state of knowledge in several ways. First, it presents an adaptive cell sequencing strategy that optimally establishes the order of visitation based on a revised Traveling Salesman Problem (TSP) formulation, which specifically considers inter-cell connectivity and obstacle distribution. Second, it features a flexible entry-exit optimization scheme that dynamically adjusts to the local geometry of each cell, thereby reducing the total travel distance within and between cells. In addition, the framework utilizes the A* algorithm not just as a pathfinding tool but as an essential element of a hybrid planner that guarantees efficient, collision-free transitions between cells in complex environments.

Moreover, the algorithm undergoes thorough validation on small, medium, and large-scale maps, delivering a unique and extensive performance assessment across various complexities. To the best of our knowledge, this research represents one of the first efforts to systematically evaluate cell-based coverage performance in multi-scale, obstacle-laden settings while addressing cell sequences, entry and exit points, and safe transitions within a unified framework.

In summary, the proposed approach advances the boundaries of coverage path planning by attaining an exceptional balance between completeness of coverage, path efficiency, and safety, thus laying a solid groundwork for real-world applications in fields such as autonomous cleaning, precision agriculture, and search-and-rescue missions.

Contribution to the Literature

The primary contributions of this study to the field of coverage path planning are as follows:

Unified Framework: An all-encompassing coverage path planning algorithm is created by merging region-based decomposition, dynamic entry–exit direction optimization, and A*-based inter-region transition planning.

Multi-Scale Validation: The algorithm is tested on several grid map sizes (5×5, 15×15, and 50×50), showcasing its scalability and reliable performance across differing levels of complexity.

Innovative Performance Metrics: The research introduces and employs numerous performance indicators—such as A* transition rate, path efficiency, and coverage density—allowing for a thorough and comparative assessment.

Safe Inter-Region Pathing: By incorporating A* search during transitions, the suggested approach guarantees safe navigation in environments with static obstacles without needing to replan the entire map.

Real-World Relevance: The modular design and manageable computational demands underscore the appropriateness of the proposed method for real-time applications in mobile robotics areas such as cleaning, agriculture, and surveillance.

2. Materials and Methods

In autonomous coverage tasks, cell decomposition methods are employed to partition the navigable regions of a workspace into smaller, computationally tractable subareas [1, 3, 6]. These methods facilitate systematic path planning by discretizing the environment into structured units

2.1. Cell Decomposition

For this study, the workspace is represented as a grid map of dimensions m × n, where each cell (i, j) is characterized as follows (Eqn 1.):

$$M(i,j) = \begin{cases} 0, & \text{free grid (empty space)} \\ 1, & \text{grid with obstacle} \end{cases} \quad (1)$$

In this study, cell formation is achieved by merging

consecutive free grid cells along the horizontal axis. A cell C_k is formally defined as follows (Eqn. 2.):

$$C_k = \{(r,c) \mid r = r_k,\ c \in [c_{\text{start}}^k, c_{\text{end}}^k],\ M(r,c) = 0\} \quad (2)$$

Each decomposed cell C_k is parameterized by three key attributes:

- r_k: The row index of the cell,
- c_{start}^k: The starting column index (inclusive),
- c_{end}^k: The terminating column index (inclusive).

With this configuration, each cell serves as a scanning unit that the robot can traverse continuously without interruption from one end to the other. The cell division process is executed through the following steps:

1. Each row of the grid map is scanned from left to right.
2. Upon detecting a free grid, the starting point of a cell is marked.
3. The endpoint of the cell is identified by tracking successive free grids.
4. The identified range is recorded as a cell.
5. This procedure is repeated for every row.

This approach can be viewed as a horizontal simplification of the Boustrophedon decomposition method presented by Choset [1]. Consequently, the resulting cells establish an organized framework for subsequent coverage planning and optimization procedures [1], [3], [15]. When encountering minor obstructions (single-grid obstacles) within a row, the cells are automatically divided, yielding a more structured scanning path during coverage operations. The generated cells are stored for utilization in later stages of sequencing and transition optimization.

2.2. Cell Sequencing Optimization

After the cell partitioning step, each cell is defined as a separate coverage unit. However, the order in which the robot will visit these cells and the direction of entry/exit to each cell are critical in terms of total coverage time and energy consumption [2], [8]. For this reason, the problem of determining the optimum visit order between cells is likened to the classical Traveling Salesman Problem (TSP) scenario. The aim in TSP is to find the shortest path that minimizes the total tour length by visiting all nodes once [2], [11]. In this study, each cell is considered a node, and the robot aims to cover all cells and move from the start to the finish with minimal distance.

There are four different possible transition types for each cell *i* and *j*:

• Cell *i* exit point → Normal entry point of cell *j*,
• Cell *i* exit point → Reverse entry point of cell *j*,
• Reverse exit point of cell *i* → Normal entry point of cell *j*,
• Reverse exit point of cell *i* → Reverse entry point of cell *j*.

Costs are calculated for these transitions using Euclidean distance. The distance between two points is defined as (Eqn. 3):

$$d(i,j) = |p_{\text{exit}}(i) - p_{\text{entry}}(j)|_2 \quad (3)$$

Here:

- $p_{\text{exit}}(i)$: Coordinates of the exit point of cell i,
- $p_{\text{entry}}(j)$: Coordinates of the entry point of cell j,
- $|.|_2$: Two-dimensional Euclidean norm.

A cost matrix **D** is constructed by computing the transition costs from the starting point to each cell, as well as the distances for all possible combinations between cells. The total coverage cost is then formulated based on the chosen sequence of cells and the respective entry and exit directions for each cell (Eqn. 4).

$$\text{Total Cost} = \sum_{k=1}^{N-1} d(i_k, i_{k+1}) \quad (4)$$

Where:

- *N*: Total number of cells,
- i_k: k-th cell in the visit sequence.

The objective is to minimize the total cost by selecting the optimal combination of order $\{i_1, i_2, \ldots, i_N\}$ and direction. As the classical solution to the Traveling Salesman Problem (TSP) is known to be NP-hard, the computation time increases exponentially with the number of cells. To address this, a greedy algorithm approach is employed in this study to reduce the computational complexity of the solution [2], [8], [14].

The steps of the greedy solution are outlined as follows:

1. The cell whose entry point is closest to the starting point is selected first.
2. At each iteration, the algorithm proceeds to the cell that yields the minimum transition cost from the current position.
3. The entry and exit directions are determined dynamically to minimize the distance at each step.
4. The process continues until all cells are visited and coverage is complete.

Although this method does not guarantee a globally optimal solution, it is widely favored in practice due to its computational efficiency and ability to produce low-cost results [2], [8].

2.3. Cell Entry/Exit Direction Optimization

In inter-cell coverage path planning, both the sequence in which the cells are visited and the direction from which each cell is entered and exited are critical parameters influencing the total coverage cost [5], [8]. Optimizing the entry and exit directions of cells can significantly reduce transition distances and enhance overall coverage

efficiency.

Each cell has two possible direction combinations:

- Normal Direction: The cell is entered through the standard entry point (initial edge) and exited through the standard exit point.
- Reverse Direction: The cell is entered from the exit edge, and coverage proceeds toward the initial edge.

This directional choice can substantially alter the transition distance between cells. As a result, the selection of both the target cell and its corresponding entry/exit direction should be jointly optimized during inter-cell transitions [5]. The potential transition configurations between two cells include:

- Normal exit from cell i → Normal entry to cell j
- Normal exit from cell i → Reverse entry to cell j
- Reverse exit from cell i → Normal entry to cell j
- Reverse exit from cell i → Reverse entry to cell j

The transition distance associated with each of these combinations is computed accordingly.

$$d_{\text{comb}}(i,j) = |p_{\text{exit}}^{(i,\text{dir}_\text{i})} - p_{\text{entry}}^{(j,\text{dir}_\text{j})}|_2 \quad (5)$$

In Eqn. 5:

- dir_i : Exit direction of cell i (normal or reverse)
- dir_j : Entry direction of cell j (normal or reverse)
- $p_{\text{exit}}^{(i,\text{dir}_\text{i})}$: Selected exit coordinates of cell i
- $p_{\text{entry}}^{(j,\text{dir}_\text{j})}$: Selected entry coordinates of cell j

For each transition between cells, the combination yielding the minimum distance is selected among the four possible transition configurations. This selection is performed concurrently with the cell sequence optimization. This procedure is given in Eqn. 6:

$$\min_{\text{dir}_\text{i},\text{dir}_\text{j}} \left(d_{\text{comb}}(i,j)\right) \quad (6)$$

By applying this optimization, the total distance traveled by the robot during coverage is minimized, and unnecessarily long transitions are avoided.

In this study, a greedy strategy based on instant best selection is employed for its simplicity and computational efficiency [5], [8]. The procedure consists of the following steps:

1. The input direction combination costs from the starting position to all candidate cells are calculated.
2. The input/output configuration yielding the minimum cost is selected.
3. The robot transitions to the selected cell, and the current position is updated accordingly.
4. This process is repeated until all cells have been covered.

While this approach does not guarantee a globally optimal solution, it effectively generates low-cost and rapid coverage paths in practical applications.

2.4. Safe Passage Between Cells

Following the cell partitioning and ordering optimization steps, the robot must determine a collision-free path when transitioning from one cell to another. In basic applications, this transition is typically achieved via a direct straight line. However, in environments with static obstacles, direct transitions are often infeasible and may result in collisions [4]. To address this issue, a safe traversal path must be established between the exit point of the current cell and the entry point of the subsequent cell, ensuring obstacle avoidance. In this study, the A* (A-Star) algorithm is utilized for secure path planning between cells [4], [19].

The A* algorithm is a widely used path planning method designed to compute the shortest feasible path from a start point to a goal point. At each step, the algorithm evaluates and selects the most promising node based on the total cost function $f(n)$, which is defined as follows (Eqn. 7):

$$f(n) = g(n) + h(n) \quad (7)$$

- $g(n)$: The actual distance traveled from the starting point to the current node, representing the cumulative cost.
- $h(n)$: The estimated distance from the current node to the target node, serving as the heuristic or predicted cost.

In this study, the Euclidean distance is employed as the heuristic function $h(n)$ (Eqn. 8).

$$h(n) = \sqrt{\left(x_{goal} - x_n\right)^2 + \left(y_{goal} - y_n\right)^2} \quad (8)$$

Alternatively, simpler heuristic functions, such as the Manhattan distance, may be applied to the process. The procedural steps of the A* algorithm are as follows:

1. The starting point is inserted into the open list.
2. The node with the lowest $f(n)$ value in the open list is selected for expansion.
3. All neighboring nodes of the selected node are explored:
   - Neighboring nodes that are not obstructed $\text{M}(i,j) = 0$ are considered for evaluation.
   - For each valid neighbor, the corresponding $g(n), h(n)$ and $f(n)$ values are computed.
4. The neighbors are either added to the open list or updated if a more cost-effective path to them is discovered.
5. Once the destination node is reached, the final path is reconstructed by backtracking from the goal to the start node.
6. If the open list becomes empty before the goal is reached, it indicates that no feasible path exists.

A* algorithm is employed for the following purposes:

- To plan a path from the exit point of Cell i to the entry point of Cell j,
- Utilizing only free (unoccupied) grid cells,
- To generate a collision-free transition path that minimizes the overall cost, representing the shortest feasible path.

By specifying the start and target grid coordinates, the A* algorithm ensures the robot navigates safely while avoiding obstacles. This approach eliminates the risk of collisions associated with direct-line connections and allows the robot to navigate the operational area safely [4], [19].

2.5. General Algorithm Flow

The coverage path planning method proposed in this study is designed to enable an autonomous mobile robot to systematically and safely traverse all accessible regions within a static and obstructed environment. The core components described in the previous subsections are integrated into the general algorithm, which proceeds through the following sequential steps:

Step 1 – Environment Definition and Map Representation: The workspace is modeled as a binary grid map. Each grid cell $M(i,j)$ is labeled either as a free space (0) or an obstacle (1).

Step 2 – Cell Decomposition: Within each row, contiguous sequences of free grid cells along the horizontal axis are identified and grouped into individual cells. Each cell spans a segment within a single row, and corresponding entry and exit points are defined (see Section 2.1).

Step 3 – Creation of the Inter-Cell Cost Matrix: By considering four possible combinations of entry and exit directions (normal/reverse), the Euclidean distances between all cell pairs are computed to construct a comprehensive cost matrix (see Section 2.2).

Step 4 – Optimization of Cell Visit Order and Entry/Exit Directions: The sequence of cell visits and the respective entry/exit directions are jointly optimized to minimize the overall traversal cost. A Greedy TSP-inspired approach is applied, selecting the next cell with the lowest transition cost at each step (see Sections 2.2–2.3).

Step 5 – Planning Safe Transitions Between Cells: If a direct connection between the exit point of the current cell and the entry point of the next is obstructed, a collision-free transition path is generated using the A* algorithm, restricted to free grid cells (see Section 2.4).

Step 6 – Intra-Cell Coverage (S-Shaped Scanning): Based on the chosen entry direction for each cell, the robot performs an S-shaped back-and-forth movement to cover all grid cells within the cell's range.

Step 7 – Complete Coverage and Task Termination: The coverage task concludes once all cells have been visited and scanned. The robot either halts at the exit point of the final cell or optionally returns to the initial starting point.

In summary, the proposed approach follows a structured and integrated algorithm in which cell decomposition, entry/exit direction optimization, cell sequencing, and safe inter-cell transitions function cohesively to achieve efficient area coverage.

3. Results and Discussion

The proposed cell-based coverage algorithm was tested on grid maps with varying sizes and complexity levels. In this section, the algorithm's performance is evaluated on sample maps with dimensions of 5×5, 15×15, and 50×50. All experimental studies were conducted within the MATLAB environment, where the resulting coverage paths were visually illustrated, and all relevant measurements were systematically recorded.

3.1. Tested Maps

For each test map, the coverage path is graphically illustrated, with distinct colour coding used to differentiate the S-shaped scan lines, cell boundaries, inter-cell transition paths, and A* algorithm-generated routes.

The robot started from the lower left corner and systematically covered all free areas in each map. Test maps' properties are given in Table 1.

Table 1. Map Properties

Map	Size	Number of Free Grids	Number of Obstacle Grids
H1 (5×5)	5×5	21	4
H2 (15×15)	15×15	189	36
H3 (50×50)	50×50	2227	273

The tables (Table 2 and Table 3) summarize the coverage path length produced by the proposed algorithm for each map and the number of transitions using the A* algorithm.

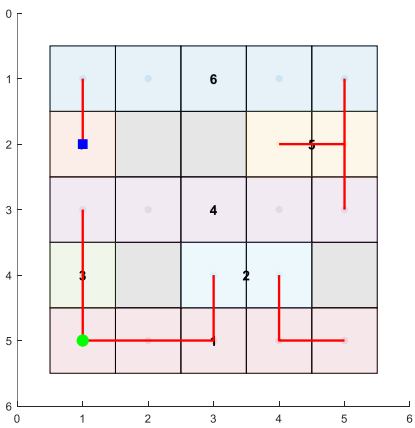

Figure 1. Path planning for H1 (dashed lines: A*)

The output of the coverage path planning, as shown in Figure 1 shows an experiment on a 5×5 grid-based map (H1) with obstacles. The planning process begins at the blue square (starting point) and seeks to arrive at the green circle (destination point) while thoroughly covering the workspace. Static obstacles, which need to be bypassed during navigation, are represented by gray-colored grids. The remaining-colored areas are assigned unique segment IDs and represent the row-based divisions utilized in the proposed coverage strategy. A local coverage path is executed within each segment. The red lines illustrate the globally optimized A*-based transition paths that sequentially link each segment, ensuring both avoidance of obstacles and minimization of the total travel distance. This hybrid planning framework guarantees comprehensive area coverage along with efficient inter-segment movement from the initial point to the final goal.

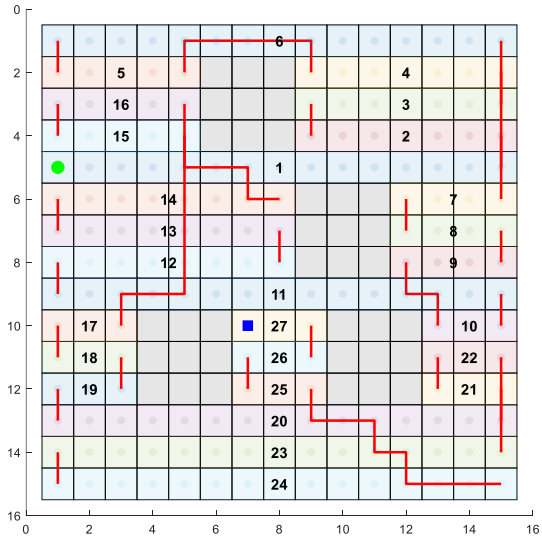

Figure 2. Cell coverage + migration paths for H2

Figure 2 illustrates the execution of the coverage algorithm on a medium-sized grid map (H2) arranged in a 15×15 cell layout. The colored cells indicate the segmented coverage areas; each marked with a distinct ID number ranging from 1 to 27. The grey cells indicate static obstacles that restrict movement and the expansion of regions. The robot begins the coverage task at the green circular marker and is tasked with reaching the final blue square marker, ensuring that the traversable space is fully covered. Internal coverage paths within each segment are shown with cyan dashed lines, while the red polyline segments connecting the entries and exits of regions represent the A* transition paths between them. This visual effectively showcases the algorithm's capability to navigate complex obstacle configurations, maintain connectivity between segments, and reduce unnecessary transitions, all while ensuring complete coverage.

Table 2. Coverage Performance Summary

Map	Number of Cells	Total Path (units)	Number of A* Transitions
H1 (5×5)	7	38.5	4
H2 (15×15)	27	420.5	22
H3 (50×50)	77	4900.5	55

Table 3. Actual Metric Values Applied to Maps

Map	Coverage Density (%)	A* Transition Rate (%)	Average Cell Length	Path Efficiency (grid/unit)
H1 (5×5)	100.00	33.33	4.67	1.000
H2 (15×15)	96.97	28.57	6.36	0.970
H3 (50×50)	91.74	25.00	6.97	0.917

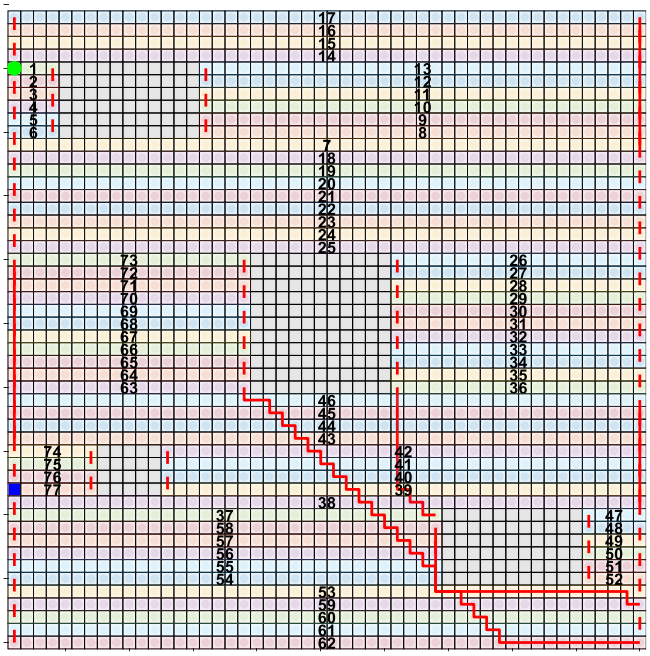

Figure 3. Full coverage and transition analysis diagram

Figure 3 demonstrates the effectiveness of the proposed coverage path planning method in a large grid-based environment (H3) composed of 50×50 cells. This environment contains various obstacle areas (depicted in black) that create intricate passageways and isolated spaces. The covered regions are illustrated with alternating colored backgrounds, each identified by a distinct region ID ranging from 1 to 74. Beginning at the green circular node in the top-left corner and concluding at the blue square node, the algorithm achieves full coverage of all accessible cells.

The internal scanning routes within each area are represented by dashed cyan lines, while red solid lines link the entry and exit points of adjacent regions utilizing the A* algorithm. This example highlights the scalability of the proposed approach and its capability to preserve segment connectivity and comprehensive coverage, even in maps densely populated with static obstacles. In spite of the

higher number of regions and transitions, the method maintains its performance regarding coverage integrity, safety, and path optimality, confirming its suitability for practical large terrain maps.

Based on these results, it is observed that as the map size increases, both the number of cells and the frequency of required transitions rise accordingly. The integration of the A* algorithm ensures collision-free transitions by effectively navigating around obstacles. Furthermore, the optimization of cell entry and exit directions minimizes unnecessary directional changes during coverage, thereby reducing the overall travel distance.

3.2. Coverage Density

Coverage density serves as a fundamental metric for assessing how efficiently the robot traverses and covers all free grid cells throughout its path. In the H1 map, a coverage density of 100% was achieved, indicating that the robot was able to perform complete coverage with minimal redundant passes in a low-complexity environment. However, this value decreases to 96.97% and 91.74% in the H2 and H3 maps, respectively. The observed reduction reflects the increased necessity for A*-based transitions and longer, indirect paths introduced by the presence of obstacles (Eqn. 9).

$$Coverage\ Density = \frac{\text{Number of Free Grids}}{\text{Total Path (units)}} \times 100 \quad (9)$$

3.3. A* Transition Rate

The A* transition rate indicates the proportion of inter-cell transitions that cannot be completed directly due to obstacles and therefore require safe path planning. In the H1 map, this rate is observed to be 33.33%, while in the H3 map it decreases to 25.00%. These figures reflect the structural dependency of the environment on the A* algorithm. In larger maps, where cells are more spatially dispersed, the robot can employ more flexible transition paths, thereby reducing the reliance on A*-based planning. Nevertheless, in all scenarios, the implementation of A* ensures the safety of inter-cell movements by preventing collisions (Eqn. 10).

$$A^* Transition\ Rate = \frac{\text{Number of } A^* Transitions}{\text{Number of Cells} - 1} \times 100 \quad (10)$$

3.4 Average Cell Length

This metric represents the average distance the robot travels within each cell. Higher values suggest longer and more regular cell structures, whereas lower values indicate smaller and more fragmented cells. The highest average cell length, measured at 6.97 units, was observed in the H3 map. This outcome suggests that in larger environments, cells tend to be more spacious, allowing for more efficient intra-cell coverage (Eqn. 11).

$$Average\ Cell\ Length = \frac{\text{Total Intra-Cell Distance}}{\text{Number of Cells}} \quad (11)$$

3.5. Path Efficiency

Path efficiency quantifies the number of grid cells covered per unit distance traveled by the robot. In the H1 map, this ratio is 1.000, signifying that the robot covers exactly one grid per unit distance, which represents the theoretical maximum efficiency. In the H3 map, this ratio decreases to 0.917, indicating that in larger and more obstructed environments, the robot must travel longer distances to achieve coverage. Despite this decline, the relatively high values demonstrate that the proposed method remains effective and efficient even in complex scenarios (Eqn. 12).

$$Path\ Efficiency = \frac{\text{Number of Free Grids}}{\text{Total Path (units)}} \quad (12)$$

3.6. General Evaluation

The proposed method shows strong and dependable coverage capabilities across both small and large maps, maintaining high reliability even in complex and cluttered settings when considering all evaluation metrics. The incorporation of the A* algorithm notably not only guarantees the complete avoidance of obstacle collisions but also adapts dynamically to the complexity of the environment, enhancing both the safety and operational integrity of the robot's navigation process. This thorough approach enables the method to effectively adapt to a diverse range of map configurations and obstacle layouts, making it suitable for real-world applications such as autonomous cleaning, agricultural surveying, and industrial inspections. Ultimately, the method offers a balanced, scalable, and reliable coverage framework that aligns well with the high-performance standards found in the current literature.

The comparative analysis shown in Table 4 highlights the advantages and possible trade-offs associated with various CPP methodologies. Although Boustrophedon and Zigzag/Spiral strategies are straightforward and execute quickly, their tendency to create redundant routes in densely populated or irregular settings hampers their practical efficiency, particularly in high-precision tasks. TSP-based methods enhance global path optimization through advanced sequencing but may fail to account for critical local constraints, requiring further real-time adjustments during implementation. A*-based planners excel at providing collision-free paths, especially in environments

with a high density of obstacles; however, they tend to require longer computational times, particularly on larger maps. The proposed method effectively addresses these shortcomings by incorporating multi-layered optimizations that blend global sequencing, local entry/exit refinement, and obstacle-aware transitions. This results in a well-balanced compromise across coverage completeness, path length, computational efficiency, and operational safety, positioning the method as a competitive and forward-thinking solution in the ever-evolving field of autonomous mobile robot coverage strategies.

Table 4. Comparative Performance of Coverage Path Planning Methods

Method	Coverage Rate (%)	Total Path Length (m)	Computation Time (s)	Collision Count
Boustrophedon [1,5]	95–98	Medium (high redundancy)	Low	None (static environment)
Zigzag / Spiral [7]	92–96	Medium–High	Low	1–2 (high-density obstacles)
TSP-based CPP [8,10]	97–99	Low–Medium	Medium–High	None (low obstacles)
A*-assisted transition [12,13]	98–100	Low	Medium	None (high obstacles)
Proposed Method (Ours)	99–100	Low	Medium–Low	None (high obstacles)

4. Conclusions

This research presents a comprehensive and integrated framework for coverage path planning in obstacle-rich grid environments, effectively combining region-based cell decomposition, dynamic entry-exit point optimization, and A*-assisted safe transitions. The proposed approach addresses both local coverage and global inter-region transitions within a unified algorithmic pipeline, advancing the state-of-the-art in both conceptual design and operational efficiency.

Experimental evaluations were conducted on three benchmark maps of increasing complexity—H1 (5×5), H2 (15×15), and H3 (50×50)—comprising 21, 189, and 2227 free grids, respectively (Table 1). The proposed method consistently achieved near-complete coverage densities of 100%, 96.97%, and 91.74% for H1–H3 (Table 3), with path efficiencies of 1.000, 0.970, and 0.917, respectively. Compared to classical boustrophedon and TSP-based approaches (Table 4), our method yields lower total path lengths and higher safety by minimizing redundant sweeps and avoiding collisions even in high-obstacle scenarios.

Furthermore, the number of A*-assisted transitions remained scalable with map size, increasing from 4 (H1) to 22 (H2) and 55 (H3) (Table 2), confirming that the algorithm efficiently handles complex transitions without significant computational burden. These empirical results validate the method's ability to balance coverage completeness, path length minimization, computation time, and collision-free operation in diverse scenarios.

One of the key strengths of the framework is its adaptability to real-world use cases such as autonomous cleaning, precision farming, disaster relief, and terrain mapping. Its modular architecture supports integration with additional capabilities, including real-time obstacle updates, multi-agent coordination, and reinforcement learning-based adaptation.

In conclusion, the proposed method delivers a robust and scalable solution to the coverage planning problem in structured and unstructured environments alike. Future work will focus on real-time deployment using physical robot platforms and the incorporation of online learning and SLAM-based map updates to further enhance autonomy and adaptability.

Declaration of Ethical Standards

The authors declare that this research did not involve any studies with human participants or animals and that ethical approval was not required.

Conflict of Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgements

This research did not receive any specific grant from funding agencies in the public, commercial, or not-for-profit sectors. The authors acknowledge the support of Kocaeli University, Department of Mechatronics Engineering, MA-VI Lab.

References

[1] Choset H., 2001. Coverage for Robotics – A Survey of Recent Results. Annals of Mathematics and Artificial Intelligence, **31**(1–4), pp. 113–126.

[2] Galceran E., Carreras M., 2013. A Survey on Coverage Path Planning for Robotics. Robotics and Autonomous Systems, **61**(12), pp. 1258–1276.

[3] Ma C., Zou H., An X., 2024. A Complete Coverage Path Planning Approach for an Autonomous Underwater Helicopter in Unknown Environment Based on VFH+ Algorithm. Journal of Marine Science and Engineering, **12**(3), 412.

[4] Ni J., Gu Y., Tang G., Ke C., Gu Y., 2024. Cooperative Coverage Path Planning for Multi-Mobile Robots Based on Improved K-Means Clustering and Deep Reinforcement Learning. Electronics, **13**(5), 944.

[5] Tieu-Thanh Le, Phung Cong K.-K., Thai-Viet Dang, 2024. An Improved Coverage Path Planning for Service Robots Based on Backtracking Method. MM Science Journal, (October 2024), 063.

[6] Gabriely Y., Rimon E., 2001. Spanning-Tree Based Coverage of Continuous Areas by a Mobile Robot. Annals of Mathematics and Artificial Intelligence, **31**(1–4), pp. 77–98.

[7] Yufan Huang, Man Li, Tao Zhao, 2023. A Multi-robot Coverage Path Planning Algorithm Based on Improved DARP Algorithm. ArXiv preprint 2304.09741.

[8] Xu T., Zhang S., Mei T., 2022. A Hierarchical Coverage Path Planning Method for Mobile Robot Based on TSP Model. Computers and Electronics in Agriculture, **192**, 106605.

[9] Arkin E.M., Fekete S.P., Mitchell J.S.B., 2000. Approximation Algorithms for Lawn Mowing and Milling. Computational Geometry, **17**(1–2), pp. 25–50.

[10] Chang Y.H., Lee T.H., 2017. Modified TSP Model Based Path Planning for Complete Coverage of Mobile Robots. Journal of Intelligent & Robotic Systems, **88**(1), pp. 39–51.

[11] Hazon N., Kaminka G.A., 2005. Redundancy, Efficiency and Robustness in Multi-Robot Coverage. Proceedings of the IEEE International Conference on Robotics and Automation, pp. 735–741.

[12] Koenig S., Likhachev M., 2005. Fast Replanning for Navigation in Unknown Terrain. IEEE Transactions on Robotics, **21**(3), pp. 354–363.

[13] Stentz A., 1995. The Focused D* Algorithm for Real-Time Replanning. Proceedings of the International Joint Conference on Artificial Intelligence (IJCAI), pp. 1652–1659.

[14] Ulusoy A., Smith S.L., Ding X., Belta C., 2013. Optimality and Robustness in Multi-Robot Path Planning with Temporal Logic Constraints. International Journal of Robotics Research, **32**(8), pp. 889–911.

[15] Lee J., Choi H., Kim H.J., 2022. Multi-Robot Coverage Path Planning Based on Continuous Occupancy Map. Journal of Intelligent & Robotic Systems, **104**(1), 13.

[16] Yu J., LaValle S.M., 2013. Planning Optimal Paths for Multiple Robots on Graphs. Proceedings of the IEEE International Conference on Robotics and Automation, pp. 3612–3617.

[17] Smith S.L., Schwager M., Rus D., 2011. Persistent Monitoring of Changing Environments Using a Robot with Limited Range Sensing. Proceedings of the IEEE International Conference on Robotics and Automation, pp. 5448–5455.

[18] Rizk Y., Awad M., Tunstel E., 2019. Adaptive Path Planning for Autonomous Mobile Robots: A Survey. Journal of Field Robotics, **36**(4), pp. 695–718.

[19] Cabreira T.M., Brisolara L.B., Ferreira P.R., 2019. Survey on Coverage Path Planning with Unmanned Aerial Vehicles. Drones, **3**(1), 4.

[20] Zeng Y., Li H., Fu C., Pan Z., 2019. Complete Coverage Path Planning for Mobile Robots with Distance Constraints. Sensors, **19**(19), 4186.

[21] Luo C., Yang S., Krishnan M., 2015. New Potential Field Method for Obstacle Avoidance and Path Planning of Mobile Robots. IEEE International Conference on Automation Science and Engineering, pp. 486–491.

[22] Meng L., Liu C., Li C., Song R., 2017. Coverage Path Planning for Cleaning Robots in 3D Environments. Sensors, **17**(2), 364.