IEEE Access
Multidisciplinary : Rapid Review : Open Access Journal

Received 4 May 2024, accepted 17 May 2024, date of publication 21 May 2024, date of current version 31 May 2024.

Digital Object Identifier 10.1109/ACCESS.2024.3403658

RESEARCH ARTICLE

Robust Online Multi-Robot Simultaneous Exploration and Coverage Path Planning

VISHNU G. NAIR[1], M. V. DILEEP[2], AND K. R. GURUPRASAD[3]
[1]Department of Aeronautical and Automobile Engineering, Manipal Institute of Technology, Manipal Academy of Higher Education, Manipal, Udupi, Karnataka 576104, India
[2]Department of Aerospace Engineering, Chungnam National University, Yuseong-gu, Daejeon 34131, South Korea
[3]Department of Mechanical Engineering, IIT Kanpur, Kalyanpur, Kanpur, Uttar Pradesh 208016, India

Corresponding authors: Vishnu G. Nair (vishnu.nair@manipal.edu) and K. R. Guruprasad (krgprao@iitk.ac.in)

ABSTRACT In this paper, a robust online Multi-robot Simultaneous exploration and coverage path planning problem is presented. The entire workspace is initially partitioned using a variant of Voronoi partitioning, Manhattan Voronoi, and the robots execute simultaneous exploration and coverage using Spanning Tree Coverage algorithm and cover the workspace. Once the robot(s) failure is detected the uncovered portions of the Voronoi cell of the failed robot will be shared between other eligible robots or a replacement strategy,if available, is performed. Simulation experiments within the V-rep environment is used to demonstrate and validate the performance of the proposed algorithm. Though the authors used the Spanning Tree Coverage algorithm for path planning for the purpose of demonstration, any suitable coverage algorithm may be used.

INDEX TERMS Coverage path planning, robot failure, dynamic partition boundaries, Manhattan Voronoi, multi-robot systems.

I. INTRODUCTION

In the context of mobile robots, several major applications involve tasks such as cleaning, mine-sweeping, and structural inspection. These applications require robots to execute complete coverage paths within the accessible workspace. Various recent approaches are presented in [1], [2], [3], [4], [25], [26], and [27]. However, planning such paths in multi-robot scenarios presents several challenges, including task duplication, coordination between robots, and optimal task allocation. To address these challenges, efficient task allocation between robots becomes a primary focus [5], [6], [7]. One effective approach in coverage path planning (CPP) with multiple robots is the 'divide and conquer' method. This approach divides the entire workspace into cells equal to the number of robots in the system [10], [11], [12], [13], [14], [18], [19]. Each robot is then assigned an individual cell to cover, eliminating the need for continuous communication with other robots.

Voronoi partitioning or its variants [14], [22], [24] are commonly used for dividing the workspace. However, most literature performs partitioning based on the initial positions of the robots, which may be less efficient without an optimal initial deployment scheme. Additionally, some methodologies combine exploration and coverage for multi-robot systems. After the partitioning phase, the robots begin covering their assigned regions while simultaneously exploring them. It is worth noting that these approaches often do not account for robot failures in their plans [20]. Therefore, it is crucial to design a robust strategy that can handle robot failures. This paper presents a strategy for multi-robot coverage path planning that addresses these challenges by introducing two methodologies, one is a dynamic workspace allocation method and the second one for robot replacement if such a replacement is available. The proposed method aims to reduce the inequality in workspace allocation in such scenarios.

In this paper, the partition and cover approach using Manhattan Voronoi [15] is explored. This approach converts a multi-robot coverage (MRC) problem into a set of single-robot coverage problems. The authors propose a method that divides the workspace into cells using the Manhattan Voronoi partitioning technique. However, this approach assumes that the obstacles in the workspace

The associate editor coordinating the review of this manuscript and approving it for publication was Wai-Keung Fung.

are known in advance since the partitioning process is offline. Consequently, it may not be possible to accommodate obstacles during the partitioning process, leading to non-uniform area allotment to the robots. Another algorithm called MRSimEx, presented in [20], combines online and offline coverage strategies. The robots in this algorithm simultaneously explore and cover their allotted areas by intermittently conducting exploration during the coverage process. This intermittent exploration helps reduce power consumption since the exploration sensors only need to be active during the exploration phase. However, the offline partitioning process used in MRSimEx does not account for obstacle placement unless they are known a priori. This limitation can affect the optimality of the coverage as robots may have unevenly distributed areas due to obstacles. If a large obstacle is present within a robot's allotted cell, that robot will need to cover a smaller area compared to others, resulting in reduced resource utilization. In contrast, offline MRC algorithms that employ Voronoi partitioning can handle obstacle scenarios since the obstacle placement is known beforehand. The main challenge in such algorithms is ensuring that the partitioned cells are not topologically disconnected, a problem addressed effectively in [21]. Overall, the choice of partitioning approach in multi-robot CPP algorithms can impact factors such as optimality, obstacle accommodation, resource utilization, and topological connectivity of the partitioned cells.The proposed algorithm offers several advantages over existing work in the literature. These advantages include achieving non-overlapping complete coverage, reducing the time taken to complete the coverage task, minimizing battery consumption, and exhibiting robustness to the failure or addition of robots.

The proposed algorithm optimizes the partitioning of the workspace into distinct cells designated to individual robots, using advanced Manhattan Voronoi techniques. This precise partitioning ensures that each robot is responsible for a unique area, eliminating the overlap that often occurs in less sophisticated systems. Non-overlapping coverage is crucial for efficiency as it avoids redundant passes over the same area, which not only wastes time but also consumes additional battery power. Moreover, by ensuring complete coverage, the algorithm guarantees that no part of the workspace is left unattended, an essential factor for applications like cleaning or mine-sweeping where missing an area could have serious consequences. By effectively assigning robots to specific partitions without overlap, the algorithm minimizes the total time required for complete coverage. Each robot operates independently within its designated area, optimizing its path planning and coverage pattern without the need to coordinate movements with other robots in real time. This autonomy allows all robots to operate in parallel at full efficiency, significantly speeding up the overall coverage process. This is particularly beneficial in large-scale operations where time efficiency translates directly to cost savings and operational effectiveness. Efficient path planning inherent in the algorithm contributes directly to reduced battery consumption. By eliminating redundant coverage and optimizing travel paths within each partition, robots can minimize idle running and unnecessary long-distance movements, thereby conserving energy. Furthermore, the algorithm's design allows for intermittent operation of exploration sensors only during necessary phases of the coverage process, which further helps in reducing the energy expenditure that would otherwise be required for continuous sensor operation. Also, one of the standout features of the proposed methodology is its robustness in the face of dynamic changes to the robot fleet, such as failures or the addition of new robots. The system is designed to dynamically reassign coverage tasks and redistribute workspace partitions if a robot fails, ensuring continuous operation without significant disruption. Similarly, if additional robots are introduced into the system, the algorithm can quickly integrate these resources by re-evaluating and adjusting the partitioning of the workspace. This flexibility is critical in practical scenarios where robotic systems must maintain operation despite hardware failures or when scaling operations.

The major contribution of this paper lies in advancing the state-of-the-art in multi-robot coverage path planning by addressing the under explored issues of dynamic allocation and failure robustness. We demonstrate through simulations on the V-Rep platform that our methodologies not only improve coverage efficiency but also achieve non-overlapping complete coverage, reduce the time taken to complete coverage tasks, minimize battery consumption, and exhibit robustness to the failure or addition of robots. This work significantly extends existing methodologies by optimizing resource utilization and ensuring continuous operation despite unforeseen disruptions, contributing valuable insights and tools to the field of robotic coverage path planning. The rest of the paper is organized as follows. The problem statement is provided in section II. In section III, the proposed algorithm is discussed followed by an illustrative example in section IV. The analysis of the algorithm is carried out in section V and provided results of simulation experiments in section VI. The paper is concluded with a brief summary on the contribution.

II. PROBLEM STATEMENT

In this paper, the problem at hand involves a bounded and contiguous workspace $Q \subset \mathbb{R}^2$ with known boundaries, containing n unknown obstacles represented as $O_i \subset \mathbb{R}^2$, where $i \in 0, 1, 2, \ldots, n$. The task is to cover the region $Q \setminus O$, where $O = \bigcup_{i=1}^{n} O_i$, using N robots, each equipped with a square-sized coverage tool of size D. The robots are capable of executing the MRSimExCoverage-STC algorithm [20] for covering the workspace. The coverage is resolution complete if all the free cells are visited by a robot and non-overlapping, meaning each cell is visited at most once. The coverage is achieved using approximate cellular decomposition schemes, such as the one described in [9].

In multi-robot coverage path planning, the initial allocation and partitioning of the workspace among robots are typically based on the Manhattan distance and centroidal Voronoi partitioning schemes [15], [19], [20]. While these methods effectively distribute the workspace under ideal conditions, they face significant challenges when unexpected scenarios, such as robot failures, occur. For instance, as depicted in figure1, a scenario with three robots where Robot 1 (R1) fails leads to uncovered regions within its assigned Voronoi cell. The Voronoi partition boundaries are shown in thick black lines. The shaded region represents uncovered regions in the Voronoi cell of R1. This situation necessitates a dynamic reallocation of the workspace to ensure that coverage continuity is maintained. The primary challenge here lies in the ability of the system to adaptively redistribute the workload among the remaining operational robots without causing overlaps or leaving gaps in coverage. The system must not only reassign the failed robot's area but also ensure that the reallocation does not disrupt the efficiency and path optimization of the other robots. Moreover, the need for a rapid adjustment is crucial to prevent any delay in coverage, which is particularly important in time-sensitive applications like mine-sweeping.

This paper aims to propose an enhanced methodology for the efficient reallocation of the workspace during the MRSimExCoverage process, which dynamically adjusts the partition boundaries between healthy robots to achieve uniform workspace distribution. By integrating robust mechanisms for adaptive re-partitioning, the proposed methodology seeks to ensure that all available robots are utilized to their best potential, thereby maintaining complete coverage of the workspace even in the event of robot failures. Achieving this level of adaptability and robustness enhances the overall efficiency and reliability of multi-robot systems engaged in critical tasks such as cleaning, mine-sweeping, and structural inspection. These tasks require not only complete coverage of the accessible workspace but also the generation of a comprehensive map of the area as a by-product, further underscoring the need for an effective and flexible coverage strategy.

III. PROPOSED METHODOLOGY

In this section, we introduce our proposed algorithm for the online allotment of workspace during the MRSimExCoverage [20] process, specifically designed to address scenarios such as robot failures. While MRSimExCoverage is efficient in terms of power usage, it does not guarantee uniform workspace allocation when facing dynamic changes in the robot configuration. To overcome this challenge, we adopt a "Partition and Cover" strategy using Manhattan distance-based Voronoi partitioning [15] for the initial workspace partitioning. However, during the re-partitioning stage, we implement geodesic distance-based Voronoi partitioning [21], which offers an effective distance measurement technique, particularly in the presence of obstacles.

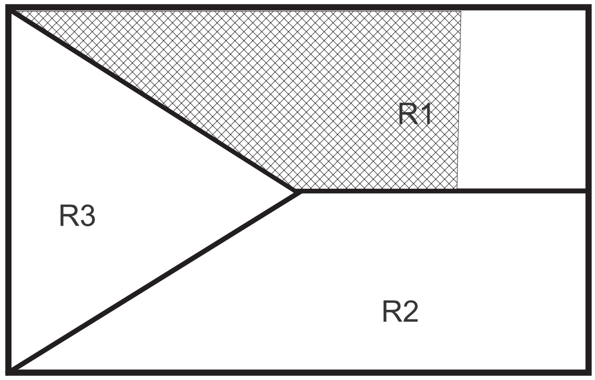


FIGURE 1. A three robot scenario in which robot 1 (R1) failed is shown. The Voronoi partition boundaries are shown in thick black lines. The shaded region represents the uncovered regions in the Voronoi cell of R1. White area represents the covered sections by individual robots.

This approach transforms the multi-robot coverage problem into individual single-robot coverage problems, leading to improved efficiency in task (workspace) allotment and ensuring uniformity in allocation. Communication requirements are kept to a minimum, with only a few messages related to robot availability, task completion, final map, robot health status, etc., exchanged between the robots. Below, we provide a concise overview of the underlying geodesic distance-based Voronoi partitioning, Manhattan distance-based Voronoi partitioning, and the MRSimExCoverage process

A. GEODESIC VPC

The underlying partitioning scheme utilized in the partitioning process is known as Geodesic distance-based Voronoi partitioning (Geodesic VPC) [21]. The geodesic distance between any two points is defined as the length of the shortest path between them while avoiding obstacles on a flat surface. In the context of mobile robots navigating on a flat surface with obstacles, the geodesic distance-based Voronoi partitioning calculates the shortest paths between points, taking into account the obstacles' presence. This approach ensures that the partitioning is based on the decomposition of the free space rather than the entire region. The geodesic distance-based Voronoi partitioning is given by [21]

$$V_i^G(\mathcal{P}) = \left\{ q \in Q \setminus O | d_G(q, p_i) \leq d_G(q, p_j), \ \forall j \in I_N \right\} \tag{1}$$

Here, $d_G(q, p)$ is the geodesic distance between points q and p and Q is the workspace being partitioned.

The advantage of the Geodesic VPC over the standard Voronoi partition using the Euclidean distance metric lies in its ability to decompose the free space, resulting in contiguous cells that are always topologically connected. This ensures that each cell assigned to the robots for coverage remains connected and accessible, even when obstacles are present. By adopting the Geodesic distance-based Voronoi partitioning scheme, our methodology guarantees more robust and efficient allocation of workspace among multiple robots, maintaining contiguity in the assigned cells, and addressing the challenges posed by obstacles in dynamic environments.

B. MANHATTAN VPC

In the MRSimExCoverage process, the underlying partitioning scheme used is the Manhattan distance-based Voronoi partitioning (Manhattan VPC) [15]. This scheme improves the efficiency of the coverage algorithm by representing the entire workspace as a union of cells with a size of $2D \times 2D$, where D corresponds to the square coverage tool footprint of the robots. By dividing the workspace into cells of size $2D \times 2D$, the coverage algorithm ensures that the robots can efficiently cover the entire area without leaving any pockets or gaps. In some cases, if this provision is not made in the coverage algorithm, robots may need to retract and restart coverage to cover left-out pockets [8]. This retracing and restarting process can lead to inefficiencies and additional time consumption. Since most of the robot motion in coverage applications predominantly occurs in horizontal or vertical directions, the use of the Manhattan distance metric in computing Voronoi cells is logical. The Manhattan distance metric considers the sum of horizontal and vertical distances between points, as opposed to the standard Euclidean distance, which considers the straight-line distance. Considering the nature of robot motion in coverage tasks, the Manhattan distance-based Voronoi partitioning proves to be more effective and efficient, leading to improved coverage results.The Manhattan distance-based Voronoi partitioning is given by [20]

$$V_i(\mathcal{P}) = \left\{ q \in Q | d_m(q, p_i) \le d_m(q, p_j), \ \forall j \in I_N \right\} \quad (2)$$

Here, $d_m(q, p)$ is the Manhattan distance between points q and p and Q is the workspace under consideration.

After the partitioning process, the robots proceed to execute single-robot coverage algorithms to cover their allotted regions. In this paper, the Spanning tree-based coverage (STC) algorithm [9] is utilized as the underlying single-robot CPP algorithm. The STC algorithm is an effective method for a single robot to efficiently cover its assigned region. It involves constructing a spanning tree over the free space of the given region. The robot then follows the edges of the spanning tree to traverse and cover all the cells in its allocated area. This approach ensures that the robot explores the entire region systematically without any overlap or omission, leading to complete coverage.

C. MRSimExCoverage PROBLEM

In the multi-robot simultaneous exploration and coverage (MRSimExCoverage) problem [20], both exploration and coverage tasks are combined to harness the benefits of both online and offline coverage algorithms. The robots are equipped with relatively longer-range sensors, which can cover the entire workspace size. The process begins with the robots generating a coverage path under the assumption that no obstacles are present, utilizing the STC algorithm. During the exploration and coverage phases, the robots activate their exploratory sensors only when they reach boundary cells between explored and unexplored regions, known as exploration windows. The intermittent exploration phases provide information to update the map, which is then used to generate the coverage path. This iterative process continues until the entire workspace is covered, leading to a complete coverage of the area, along with obtaining a map of the region as a byproduct.

However, as the robots start the coverage process using the MRSimExCoverage STC algorithm, there can be robot failure scenario which can lead to non-uniform coverage loads for the robots, as the obstacle-free regions within the Voronoi cells may differ depending on the obstacle scenario. To eliminate this issue and to achieve a more uniform workspace allocation, a portion of the uncovered Voronoi cells of the failed robot must be allotted to the remaining 'healthy' robots or replace the 'dead' robot with a new healthy one, while ensuring the contiguity of the assigned cells. The detailed procedure for achieving this more uniform workspace allocation is provided in the following section.

D. THE PROPOSED METHODOLOGY

Let us consider a multi-robot scenario consisting of N robots covering a workspace. There are some assumptions made as follows.

1) After the initial Manhattan Voronoi partitioning all the major cells are given identification numbers. (The major cells can be easily calculated provided the workspace boundaries are known.)
2) All the robots have a priori information about the total number of major cells in the workspace and its corresponding id numbers.
3) Each of the robots knows which and how many cells are allotted to itself as well as to others in the system after the initial Manhattan Voronoi partitioning.
4) Each robot separately stores the Voronoi boundary cells of itself and of others.
5) All robots send "I am alive" signal periodically to all other robots and
6) All robots send the cell numbers covered by it and occupied by obstacles detected by it periodically.

Figure 2 illustrates a scenario with three robots engaged in the MRSimExCoverage STC algorithm [20]. An obstacle free workspace is considered for simplicity. In the event of a robot failure, let's say Robot 3 (R3) fails, the remaining robots become aware of this failure due to the absence of the "I am alive" signal from R3. To address such scenarios, this paper presents two solutions:

1) **Voronoi Cell Reallocation:** This approach involves redistributing the Voronoi cells initially assigned to the failed robot among the remaining robots. This reallocation ensures a more balanced coverage distribution. The pseudo-code detailing the steps for this process for

each robot is provided in algorithm 4. Eventhough the major process here is repartitioning and re allocating the major cells of Voronoi cells of the failed robot, the underlying algorithms are 1,2 and 3. Let us first explain these algorithms and then proceed to algorithm 4.

Initially, the workspace, denoted as ‘Q’, is partitioned into areas assigned to each robot using a Manhattan Voronoi partitioning method based on the robots’ initial positions. This strategic division ensures that each robot has a clearly defined area to monitor or maintain, facilitating efficient and non-overlapping coverage. Once the initial setup is complete, the system enters a continuous operational loop, indicating it’s designed for tasks requiring uninterrupted monitoring, such as surveillance or environmental management. Each cycle of this loop begins with reinitializing the robots’ positions, workspace boundaries, and communication channels to ensure all elements of the system are perfectly synchronized for the tasks ahead.

During active coverage, each robot performs its duties within its assigned area using a method referred to as MRSimEx Spanning Tree Coverage (STC) (algorithm 3). The MRSimex STC process begins with exploration, where each robot systematically surveys its environment to identify obstacles, free spaces, and unknown areas(algorithm 2). The initial step involves scanning the environment using the sensor, which provides a comprehensive view of the surroundings in all directions. This scan allows the robot to distinguish between areas that are occupied by objects or obstacles and those that are free and unobstructed. Upon completing the scan, the robot updates multiple lists based on its observations. The ‘occupied cell’ list is updated with information about areas identified as being occupied by objects or obstacles. This list serves as a record of the locations of obstacles within the environment, enabling the robot to navigate around them effectively. Conversely, the ‘free cells’ list is updated with information about areas identified as being free and unoccupied. These areas are essential for the robot’s navigation and movement through the environment, as they provide clear pathways for the robot to traverse without encountering obstacles. Additionally, the algorithm maintains an ‘unknown cell’ list, which includes areas where the sensor data is inconclusive or ambiguous. These areas may require further investigation or verification to determine their true status, ensuring that the robot’s map of the environment remains accurate and up-to-date. Finally, the ‘frontier cell’ list is updated, which comprises the boundary between known and unknown areas. These frontier cells represent areas adjacent to both occupied and free spaces and are prioritized for further exploration or observation. By updating this list, the robot can systematically expand its knowledge of the environment, ensuring thorough coverage and accurate mapping of the area.This initial exploration phase allows the robot to gather data about its surroundings, creating a foundational map of the workspace.

Following exploration, the robot generates a Spanning Tree (ST) over the explored and not-covered ‘free’ major cells. This ST likely serves as a roadmap for the robot’s subsequent coverage path, guiding its movements through the free areas of the workspace while ensuring comprehensive coverage. Once the ST is established, the robot generates a Coverage Path (CP) through sub-cells, circumnavigating the edges of the ST on their right side. This CP is designed to systematically traverse the free major cells within the robot’s assigned area, ensuring thorough coverage while minimizing redundant movements. Throughout the exploration and coverage process, the robot periodically checks for an ‘exploration window’, likely a predefined area or condition that prompts the robot to reinitiate the exploration phase. This periodic reassessment ensures that the robot continues to update its map and adapt its coverage strategy as needed.

Finally, the robot terminates its coverage path and exploration efforts when it returns to the starting sub-cell, signaling the completion of its assigned task. This systematic approach enables each robot to autonomously partition its designated area, explore its surroundings, and generate a comprehensive coverage path, contributing to the overall efficiency and effectiveness of the multi-robot system in achieving complete coverage of the workspace. The MRSimEx STC is approach maximizes area coverage while minimizing overlaps and gaps. To maintain system integrity, robots periodically send an “I am alive” signal to confirm their operational status, enhancing the system’s fault tolerance by enabling early detection of any robot failures. If a robot fails, the system is designed to respond immediately: the failing robot sends a “Hault” message to its neighbors, likely to prevent any actions that might interfere with reallocation efforts. From here, the system can either redistribute the failed robot’s area among the remaining functional robots or replace the failed robot if a spare is available. This dual-option recovery mechanism ensures that the coverage task continues smoothly without significant interruptions, thereby maintaining overall system efficiency and reliability.

If during robot failures, the Voronoi cell reallocation is chosen as per1, then the algorithm 4 will be executed. Initially, the “RESTRUCTUREVORONOI” procedure is invoked to adjust the Voronoi boundaries based on specific criteria. This involves calculating Manhattan distances from uncovered cells to Robots 1 and 2 within Robot 3’s Voronoi cell, utilizing the “CALCULATEMANHATTANDISTANCES” procedure. These distances inform the adjustment of Voronoi boundaries via the “ADJUSTVORONOIBOUNDARIES” procedure, ensuring an optimized allocation of cells among the robots. Subsequently, the "ALLOTMAJORCELLS" procedure allocates major cells to each robot according to predefined rules. Uncovered cells, excluding those already covered by major cells, are prioritized, while unknown cells not occupied by known obstacles are considered. The allocation prioritizes cells close to Voronoi boundaries, avoids coverage overlap, allocates common boundary cells, and favors cells at the intersection of Voronoi boundaries, ensuring comprehensive coverage without redundancy or gaps. Once the Voronoi boundaries are adjusted and major cells allocated, coverage resumes seamlessly using the MRSimEx Coverage STC algorithm through the "RESUMECOVERAGE" procedure. Concurrently, unnecessary coverage tools and sensors are switched off for each robot to conserve energy, facilitated by the "SWITCHOFFCOVERAGETOOLS" procedure. For navigation within the allocated cells, the "EXECUTEMANHATTANPATH" procedure executes a geodesic Manhattan path, allowing robots to efficiently traverse the workspace while adhering to the defined Voronoi boundaries. This systematic approach ensures optimized coverage allocation, efficient resource utilization, and coordinated navigation within the multi-robot system, ultimately enhancing overall coverage efficiency and effectiveness.

2) **Replacement of the Failed Robot:** In this solution, a new robot is introduced to replace the failed one. This new robot takes over the coverage responsibilities of the failed robot. The pseudo-code outlining this process for each robot is presented in algorithm 5. Initially, the "REPLACEFAILEDROBOT" procedure is invoked upon detection of a failed robot, with its cell location provided as input. The procedure begins by identifying a minor cell within the failed robot's major cell, achieved through the "GETMINORCELL" procedure. This minor cell serves as the starting point for the replacement robot. Subsequently, a new robot is initialized at the minor cell location obtained in the previous step. The replacement robot is then directed to move to the cell previously occupied by the failed robot, facilitated by the "MOVETOCELL" procedure. Throughout this process, the replacement robot transmits an "I am alive" signal to all other robots, ensuring continuous communication and coordination within the system via the "TRANSMITSIGNAL" procedure. Additionally, the replacement robot shares Voronoi cell data obtained from the failed robot with the other robots, enabling them to update their maps and adapt their coverage strategies accordingly. This data sharing is crucial for maintaining consistency and accuracy within the multi-robot system and is achieved through the "SHAREVORONOICELLDATA" procedure. Once the replacement robot is in position and equipped with the necessary information, the coverage process resumes from the designated cell, ensuring uninterrupted coverage operations within the workspace. This seamless transition from the failed robot to its replacement ensures operational resilience and continuity within the multi-robot system, ultimately enhancing overall efficiency and reliability.

These proposed solutions enable the multi-robot system to adapt and efficiently manage scenarios of robot failure, ensuring uninterrupted coverage of the workspace.

1) VORONOI CELL REALLOCATION

In the scenario of a robot failure within the system(see figure 2), the ongoing coverage process comes to a halt. The healthy remaining robots then assume responsibility for the uncovered and unoccupied cells that were originally assigned to the failed robot. To address this challenge, the following steps are executed:

- **Calculation of Manhattan Distances:** The Manhattan distances are calculated from the uncovered cells within Robot 3's Voronoi cell to the current positions of Robots 2 and 1.
- **Restructuring of Voronoi Boundaries:** Based on these calculated distances, the Voronoi boundaries are adjusted and restructured. The dotted lines in figure 2 represents the new Voronoi boundaries.

Let us now set some rules for the major cell allotment for the i-th robot

1) **Uncovered Cells:** All known, uncovered cells that were initially assigned to the failed robot(s) will be considered. This excludes covered major cells to prevent coverage overlap.
2) **Unknown Cells:** All unknown cells belonging to the failed robot(s) will be taken into account. However, cells occupied by known obstacles will not be allotted.
3) **Boundary Proximity:** Cells situated close to the boundaries of the Voronoi cells are given priority for allotment.
4) **Avoiding Coverage Overlap:** Cells that might lead to coverage overlap, if they are unknown, will be disregarded.
5) **Common Boundary Cells:** Only the major cells positioned on the common boundary of the Voronoi cells, between the i-th robot and its neighboring robots, will be allocated.
6) **Intersection Cells:** Cells located at the intersection of two boundaries of the Voronoi cells are preferred for allocation, provided all preceding conditions are met.

Upon the completion of this reallocation process, the robots will seamlessly resume the coverage of the region using the MRSimEx Coverage STC algorithm. There can be situations where the robot need to pass through the already covered cells of the failed robot. If such a situation arises, the robot will switch off its coverage tools and sensors. Now there are two options. It can either use Manhattan path or can follow the same path of R1 to reach the next closest uncovered cell in it's tree. But the second option is time consuming as well as the battery usage will be more since the number of turns and

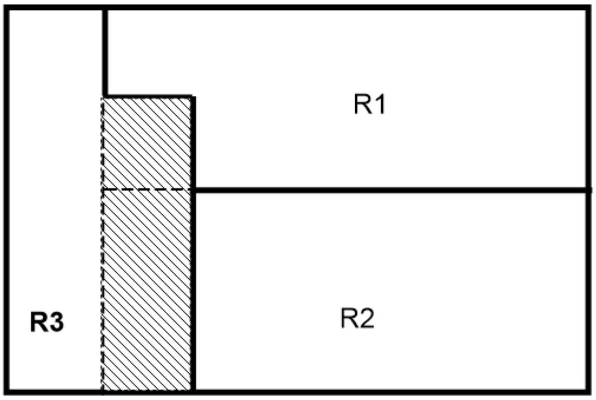


FIGURE 2. The white regions represent the areas that have already been successfully covered, while obstacles are omitted for simplicity. The Voronoi cell boundaries are shown in thick black lines. The shaded regions within Robot3's Voronoi cell indicate the uncovered regions due to the failure of R3.The newly assigned boundaries after the proposed methodology is shown in dotted lines.

the path length is more.So most preferably the first option will be chosen. A practical example that illustrates this approach is provided in section IV-A.

2) REPLACEMENT OF THE FAILED ROBOT

In this solution, when a robot experiences a failure, it is promptly replaced by a new, fully operational robot. Given that the cell number assigned to the failed robot is known, the new robot can navigate precisely to the same cell using a geodesic Manhattan path [18]. Upon arriving at this designated cell, the new robot initiates communication by transmitting an "I am alive" signal to all other robots. Simultaneously, the remaining robots share data regarding the Voronoi cell of the failed robot (R3) with the new replacement robot (R3new). Once this information exchange is completed, the coverage process resumes, ensuring seamless continuity. The new robot can be initially positioned anywhere close to the Voronoi cell of the failed robot. However, it's advisable to place this new robot, at the minor cell of the initial major cell of the failed one, positioned on the opposite side of the spanning tree. By employing this strategy, the new robot can initiate coverage through the previously unexplored side of the spanning tree, circumnavigating it on the other side. A practical example that illustrates this approach is provided in section IV-B.

IV. ILLUSTRATIVE EXAMPLE

An illustrative example of the proposed allocation methodology is given in this section. Voronoi cell reallocation is illustrated initially followed by replacement of failed robot scenario.

A. VORONOI CELL REALLOCATION

Let's delve into the scenario depicted in figure 4, which encompasses an area comprising 100 major cells, each measuring 2D $\times$ 2D in size. This notion of 2D $\times$ 2D cells is extensively elaborated in references such as [9] and [20], and a representation is available in figure 3. The cells marked with dotted boundaries represent sub-cells with corresponding sub-nodes indicated as black diamonds. Failure to integrate such a provision in the coverage algorithm can compel the robot to withdraw and recommence coverage to cover the leftover pockets, as demonstrated in [8]. In the setting of figure 4, the task of coverage involves three robots, namely R1, R2, and R3. The rectangular obstacle occupies six major cells. The shaded black region illustrates the segment of the obstacle known to R1 after the first exploration. This knowledge reduces the obstacle-free area to 94 major cells.

Initial Voronoi partitioning is executed based on the Manhattan distance metric, indicated by bold lines. Out of the 100 major cells, 34 are assigned to R1, while both R2 and R3 each receive 33 cells. This allocation is

Algorithm 1 MRsimx STC in Robot Failure Scenario

Partition Q into $V_{2Dmi}(\mathcal{P}(0))$.
While 1 **do**
  initialize robots' positions, workspace, and communication channels
  generate initial Manhattan Voronoi partitioning
  assign major cell IDs to robots based on initial partitioning
  MRSimEx STC coverage
  Send "I am alive" signal to other robots once in 5 minutes.
  if Robot failure detected **then**
    Send "Hault" msg to Neighbors
    Choose reallocateMajorCells or
    Choose replaceFailedRobot

Algorithm 2 Explore

Scan 360^o sensor
Identify occupied/free space.
Update the 'occupied cell' list
Update the 'free cells' list
Update the 'unknown cell' list
Update the 'frontier cell' list.

Algorithm 3 MR-SimExCoverage-STC

Partition Q into $V_{2Dmi}(\mathcal{P}(0))$ ($\mathcal{P}(0)$. Each robot follows Steps:
Explore.
Generate ST over the explored and not covered 'free' major cells.
Generate a CP through sub cells circumnavigating the ST edges on their right side.
When the robot reaches an 'exploration window' GOTO Explore.
If the starting sub cell is reached - STOP.

Algorithm 4 reallocateMajorCells

```
procedure RESTRUCTUREVORONOI(uncovered_cells, unknown_cells, known_obstacles)
    manhattan_distances ← CALCULATEMANHATTANDISTANCES(uncovered_cells)
    ADJUSTVORONOIBOUNDARIES(Manhattandistance)
    ALLOTMAJORCELLS(uncovered, known_obstacles)
end procedure
procedure CALCULATEMANHATTANDISTANCES()uncovered_cells
    Calculate Manhattan distances to Robots 1 and 2 from uncovered cells in Robot 3's Voronoi cell
end procedure
procedure ADJUSTVORONOIBOUNDARIES()manhattan_distances
    Adjust Voronoi boundaries based on calculated Manhattan distances
end procedure
procedure ALLOTMAJORCELLS(uncovered_cells, unknown_cells, known_obstacles)
    Allot major cells according to the specified rules:
        Rule 1: Consider uncovered cells excluding covered major cells
        Rule 2: Consider unknown cells excluding cells occupied by known obstacles
        Rule 3: Prioritize cells close to Voronoi boundaries
        Rule 4: Disregard cells that might lead to coverage overlap
        Rule 5: Allocate common boundary cells
        Rule 6: Prefer cells at the intersection of two Voronoi boundaries
end procedure
procedure RESUMECOVERAGE(robots)
    Seamlessly resume coverage using MRSimEx Coverage STC algorithm
end procedure
procedure SWITCHOFFCOVERAGETOOLS(robot)
    Switch off coverage tools and sensors for the robot
end procedure
procedure EXECUTEMANHATTANPATH(robot)
    Execute geodesic Manhattan path to reach the next cell
end procedure
```

Algorithm 5 replaceFailedRobot

```
procedure REPLACEFAILEDROBOT(failed_robot_cell)
    new_robot_cell ← GETMINORCELL(failed_robot_cell)
    new_robot ← ROBOT(new_robot_cell)
    MOVETOCELL(new_robot, failed_robot_cell)
    TRANSMITSIGNAL(new_robot)
    SHAREVORONOICELLDATA(new_robot, failed_robot_cell)
    RESUMECOVERAGE(new_robot)
end procedure
procedure GETMINORCELL(major_cell)
    Calculate and return the minor cell of the given major cell
end procedure
procedure MOVETOCELL(robot, target_cell)
    Move the robot to the specified cell using a geodesic Manhattan path
end procedure
procedure TRANSMITSIGNAL(robot)
    Transmit "I am alive" signal to all other robots
end procedure
procedure SHAREVORONOICELLDATA(robot, failed_robot_cell)
    Share Voronoi cell data of the failed robot with the new replacement robot
end procedure
procedure RESUMECOVERAGE(robot)
    Resume coverage process from the designated cell
end procedure
```

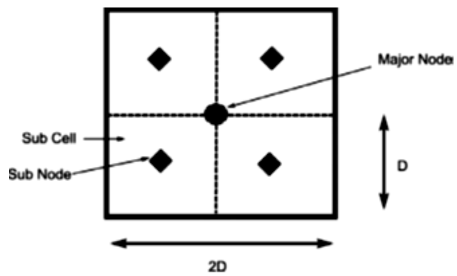


FIGURE 3. A major cell with sub-cells [9], [20].

performed without considering obstacle-related information during the partitioning phase. An approach to circumvent this situation and optimally assign cells to each robot is proposed in [23]. The shaded grey region signifies the unexplored area due to the shadow cast by the obstacle. The robots initiate the MRSimEx Coverage STC algorithm [20], commencing the process of covering their respective Voronoi cells. In this context, the interplay between cell allocation, obstacle awareness, and coverage algorithms becomes pivotal in ensuring efficient and comprehensive workspace coverage, as demonstrated in this illustrative scenario.

Let's now consider the scenario where Robot R1 experiences a failure. The location of R1 at the time of the failure is depicted as a red disc in figure 5, while the grey cells represent the concurrent locations of R2 and R3. The ongoing coverage process comes to a halt, initiating the process of reallocating major cells.

The major cells located within the Voronoi cell of R1 are numbered for reference. The following categorization is applied to these cells:

- Cells from one to nine are unexplored, unknown, and uncovered by R1.
- Cells numbered from 10 to 14, as well as 19, 22, and 25, are known and explored by R1 but remain uncovered.
- Cells numbered from 15 to 18, along with 20, 21, 23, 24, and 26 to 30, are partially covered. This coverage is partial due to not all minor cells within these major cells being covered.

This comprehensive classification of major cells accounts for their exploration, coverage status, and the specific state of each cell within the context of Robot R1.

The geodesic Manhattan distance [18] is determined from each of the major cell nodes to the current positions of both R2 and R3. The closest cells based on this distance calculation are then allocated to these robots. All cells, excluding those that are completely covered or occupied by obstacles (i.e.,

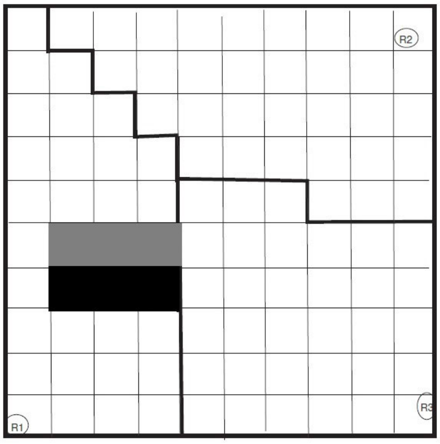


FIGURE 4. **The Manhattan Voronoi partitioned workspace encompasses an area comprising 100 major cells, each measuring 2D × 2D in size. The shaded black region illustrates the segment of the obstacle known to R1 after the first exploration. The shaded grey region signifies the unexplored area due to obstacle shadow.**

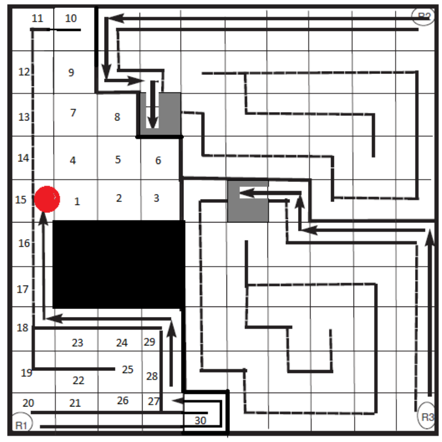


FIGURE 5. **R1 failure scenario. The location of R1 at the time of the failure is depicted as a red disc while the grey cells represents the locations of R2 and R3.**

known cells), are taken into account for reallocation. This process abides by the rules outlined in Section III-D1. The resulting reallocated partitioning boundary is indicated by the thick blue color line in figure 6 (a). Specifically, cells numbered 4 to 14 are allocated to R2, while cells 15 to 30 are assigned to R3. This reallocation process ensures that the remaining robots are assigned major cells in a strategic manner, taking into account both the proximity and the specific criteria for cell allocation.

After the completion of the partitioning process, the robots promptly resume the execution of the MRSimEx Coverage STC algorithm [20]. Spanning trees are regenerated for each robot based on their current locations and are subsequently merged with the existing ones. The newly generated spanning trees are visually represented as thick green lines in figure 6(b). The merger of the newly generated spanning trees with the existing ones enables seamless coverage continuity in the dynamically adjusted workspace allocation.

Subsequent to the reallocation process, both robots, R2 and R3, recommence their coverage operations. The path of their movements is depicted as thick arrowed lines in figure 6(c). In this illustrative example, the positions have been chosen in a way that facilitates a smooth coverage path for R2. However, the case of R3 presents a more intricate situation. As R3 resumes its coverage using the STC algorithm, it follows a path akin to circumnavigating the spanning tree, similar to R2. Eventually, R3 reaches a cell that coincides with the initial cell of R1, represented as a blue disc in figure 6(c).

Here, two options emerge to circumvent the issue of coverage overlap or addressing already covered cells. The first option involves utilizing the Manhattan distance metric to navigate to the next closest uncovered cell in its spanning tree. The second option entails deactivating its coverage tools and sensors and following the path taken by R1 to reach the next closest uncovered cell in its spanning tree. Although the second option is more time-consuming and consumes more battery due to increased path length and turns, R3 opts for the first option. It executes a Manhattan path to reach the next closest uncovered cell in its spanning tree. This choice optimizes both time and energy resources. The path taken by R3 in this scenario is depicted as a red arrowed line in figure 6(c). Upon reaching the uncovered cell within its spanning tree, R3 resumes its coverage operations. The final scenario, as depicted in figure 6(d), portrays the successful exploration and coverage of the entire workspace. This example aptly illustrates how the proposed methodology not only manages reallocation and robot failures but also optimizes coverage paths and resource usage, ultimately leading to comprehensive and efficient coverage of the workspace.

B. REPLACEMENT OF THE FAILED ROBOT

Let's consider a scenario analogous to the depiction in figure 5. In this situation, the new replacement robot can be initially positioned anywhere close to the Voronoi cell of R1. However, it's advisable to place this new robot, referred to as R1new, at the minor cell of the initial major cell of R1, positioned on the opposite side of the spanning tree.

By employing this strategy, R1new can initiate coverage through the previously unexplored side of the spanning tree, circumnavigating it on the other side. As R1new reaches the current major cell location of the failed R1, exploration is conducted, and a new spanning tree is generated. This new tree, indicated by dashed lines, is then connected with the existing tree structure of the failed R1. A portion of the coverage path of R1new is illustrated in figure 7(a).

This iterative process continues until R1new reaches the final covered minor cell of R1, which is the exact location where R1 initially failed. This approach guarantees that coverage overlap is entirely eliminated. The final scenario,

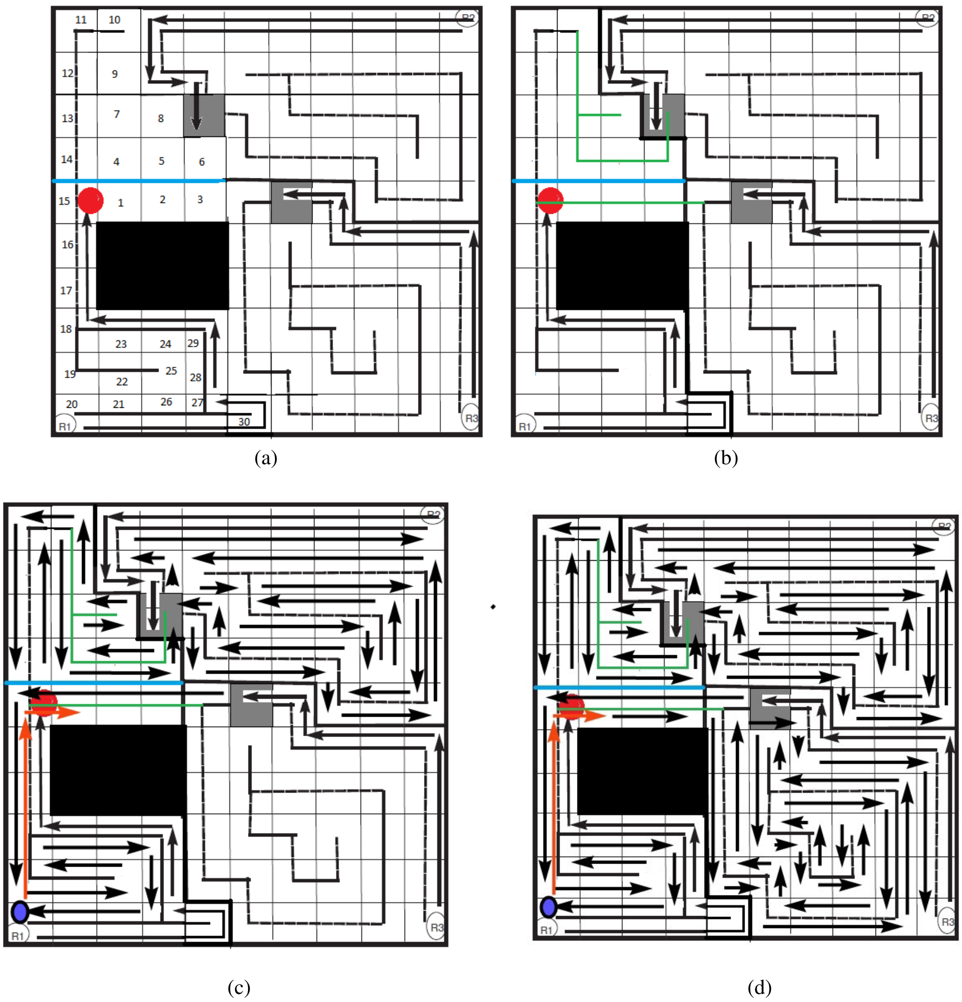


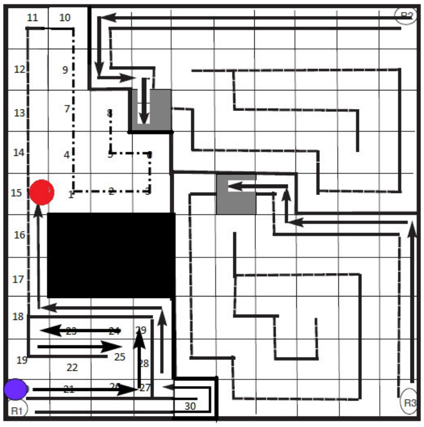
FIGURE 6. **Reallocation of partitioning boundaries. a) Reallocated partitioning boundary indicated by thick blue color. Cells 4 to 14 are allocated to R2, while cells 15 to 30 are assigned to R3. b) The newly generated spanning trees are represented as thick green lines. c) R3 reaches a cell that coincides with the initial cell of R1, represented as a blue disc. The path taken by R3 is depicted as a red arrowed line. d) Final Scenario. The dotted line shows the spanning tree generated and the arrowed lines represent the path of the robot.**

depicted in figure 7 (b), portrays the successful exploration and coverage of the entire workspace.This strategy ensures efficient coverage continuation while mitigating redundancy and overlap issues, thus maximizing the efficiency and effectiveness of the multi-robot coverage process.

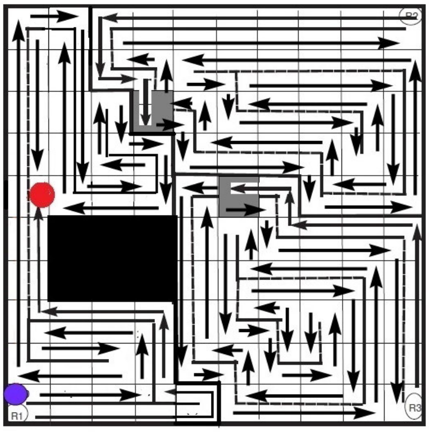
V. ANALYSIS OF THE PROPOSED ALGORITHM

In this section, an exploration of several properties inherent to the proposed algorithm is presented. A key characteristic of the algorithm lies in its dynamic nature, particularly in terms of Voronoi partitioning, which is recalculated whenever robot failures are detected. This dynamic partitioning approach ensures adaptability to changing conditions and the accurate allocation of tasks among robots. The following assumptions are made.

1) The assumption that each of the initial Voronoi cells, generated within a 2D $\times$ 2D workspace using the Manhattan distance metric, is contiguous may not hold in all practical scenarios. Situations can arise where topologically disconnected Voronoi cells result from the initial partitioning. Handling such cases necessitates approaches provided in references such as [16], [17], and [20].
2) In certain practical scenarios, the shape of the allocated Voronoi cells may require certain robots to execute more turns, potentially resulting in higher battery consumption compared to moving along straightforward paths. It's important to underline that in this paper, the consideration of battery power required for turning is excluded. Furthermore, it is assumed that the time taken by the robots to cover a workspace using the STC algorithm depends solely on the distance covered and not on the number of turns performed
3) Spanning tree edges can be regenerated in scenarios where some of the major cells, previously involved in the creation of spanning tree edges, are reassigned



(a)



(b)

FIGURE 7. a) The new spanning tree generated by R1new, shown as dashed lines. b) Final Scenario. The dotted line shows the spanning tree generated and the arrowed lines represent the path of the robot.

to neighboring robots. This reassignment occurs when major cells are allotted to different robots due to reallocation or other dynamic adjustments.

A. CHARACTERISTICS OF THE PROPOSED ALGORITHM

We list a few characteristics of the proposed algorithm without formal proof.

- The algorithm is designed to handle robot failures effectively. When a robot fails, it either reallocates its unexplored and uncovered cells among the remaining robots or replaces the failed robot with a new one.
- The algorithm ensures seamless continuity of coverage operations after a robot failure. This is achieved by redistributing tasks or introducing a new robot that takes over the responsibilities of the failed robot.
- The algorithm dynamically adapts to changes in the workspace due to robot failures. It recalculates Voronoi partitions or introduces new robots to maintain efficient coverage.
- It minimizes coverage overlap by considering previously covered cells and using geodesic Manhattan paths to navigate to the nearest uncovered cells.
- Robustness to robot failures can be easily included without much changes in the algorithm.
- The algorithm takes into account battery consumption by considering the reduction in exploration instances, especially in comparison to online algorithms. It also aims to reduce exploration time through reallocation, potentially leading to lower battery usage.
- It aims for more uniform task allocation, distributing major cells as evenly as possible among the functioning robots, enhancing overall efficiency.
- In cases where reallocation is necessary, the algorithm may take longer to complete coverage compared to the original algorithm (MRSimExCoverage STC). However, it still offers advantages over some other algorithms, such as MSTC and MFC, in terms of completion time.
- The algorithm is applicable in practical scenarios, such as office-like environments, where robots need to explore and cover areas efficiently.
- The algorithm can handle obstacles in the workspace by excluding obstacle-occupied cells from reallocation, ensuring efficient coverage around obstacles.

VI. RESULTS AND DISCUSSIONS

In this section, we present the outcomes of simulation experiments conducted within the V-Rep simulation environment to showcase the efficacy of the proposed algorithm. The simulation employs a differential-wheeled DR12 robot model, equipped with an exploration sensor, and assumes that the robot possesses localization capabilities. Localization can be achieved through various techniques, including the use of Bluetooth, gyroscopes, odometry, or algorithmic methods like SLAM (Simultaneous Localization and Mapping). For the purposes of this study, we simulate an office-like environment, and the experiments involve five robots. The multi-robot simulation begins with the initialization of the digital workspace, where each robot is assigned an initial position and the boundaries of the environment are defined. Utilizing the Manhattan Voronoi partitioning method, the workspace is then divided into distinct areas allocated to each robot. This strategic division ensures that every robot has a clearly defined region to monitor within the simulation environment, facilitating efficient and non-overlapping coverage. Operating in a continuous loop, the simulation replicates tasks requiring uninterrupted monitoring, such as surveillance or environmental management. At the start of each cycle, the positions of robots, workspace boundaries, and communication channels are reset to synchronize all elements of the system, simulating real-time operational conditions. Within the simulation, each robot autonomously explores its designated area using the MRSimEx Spanning Tree Coverage (STC) algorithm. Beginning with systematic environmental scans using simulated sensors, the robot

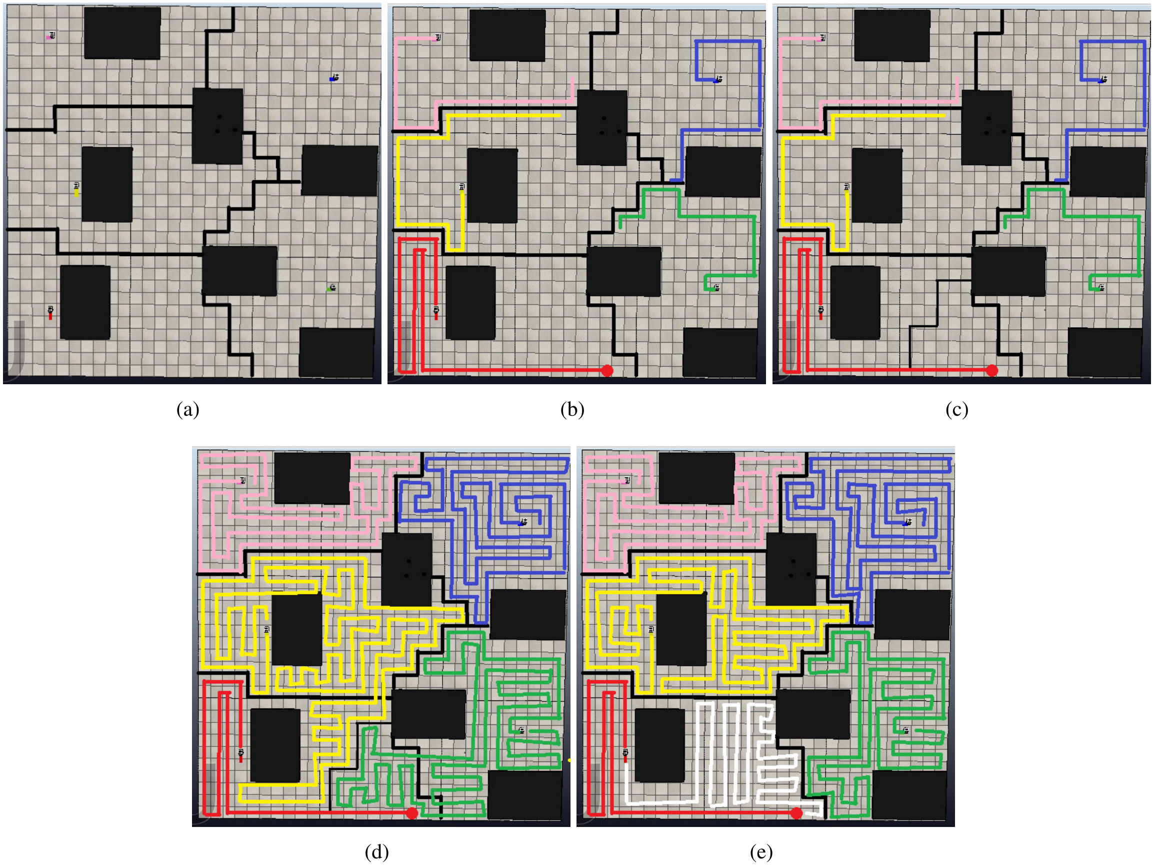


FIGURE 8. Snapshots of various stages of coverage with five DR2 robots in V-rep simulation environment. Obstacles are shown with black rectangles, black thick lines represent the partition boundaries and the coloured lines represents robot path. (a) Initial partitioning. (b)The robots starts the coverage process. One of the robot (red) failed after some time and the location of the same at that instant is shown as a red disc. (c) New partitioning. The uncovered cells are allotted to both yellow and green robots. New partitioning boundary is shown in thin black line.(d) Final scenario after Voronoi cell reallocation methodology. (e) Final scenario after robot replacement methodology. White lines shows the path of the newly added robot.

gathers data on obstacles, free spaces, and unknown areas. This data is then used to update lists of occupied, free, unknown, and frontier cells, establishing a foundational map of the workspace. Using the generated data, the simulation constructs a Spanning Tree (ST) over unexplored and not-covered free major cells, serving as a roadmap for subsequent coverage paths. The robot then generates a Coverage Path (CP) through sub-cells, systematically traversing free major cells to ensure comprehensive coverage while minimizing redundant movements. To simulate real-world scenarios, the simulation includes mechanisms for fault tolerance and recovery. Periodic "I am alive" signals confirm the operational status of robots, enabling early detection of failures. In the event of a robot failure, simulated actions are taken based on proposed solutions: Voronoi Cell Reallocation or Replacement of the Failed Robot. Simulated algorithms redistribute uncovered cells of failed robots among remaining robots or introduce a new simulated robot to replace the failed one. These actions are guided by rules governing the allotment of major cells to ensure comprehensive coverage without redundancy or gaps, prioritizing uncovered cells, avoiding overlap, and favoring boundary and intersection cells for allocation among robots. After reallocation or replacement, simulated robots seamlessly resume coverage, conserving simulated energy and ensuring efficient navigation within allocated cells

Figures 8(a)-(e) depict various stages of exploration and robot coverage paths, accompanied by the updated Voronoi cell boundaries as the robots failure is detected. Initially, the workspace is partitioned, visualized by black thick lines representing partition boundaries. Obstacles are denoted by black rectangles, while colored lines illustrate robot paths. As the coverage process commences, robots dynamically navigate through their assigned areas, systematically exploring and mapping the environment. As time progresses, a scenario unfolds where one of the robots, depicted in red, experiences a failure. At this point, the simulation marks the location of the failed robot with a red disc, signaling the interruption in the coverage process. To address this challenge, the simulation invokes the Voronoi cell reallocation methodology. In the subsequent stage, marked by a new partitioning configuration, uncovered cells are redistributed among the remaining robots, indicated by yellow and green. The new partitioning boundary is depicted by a thin black line, reflecting the adjusted allocation of cells to ensure comprehensive coverage. Following the Voronoi cell reallocation, the simulation reaches its final scenario, where coverage resumes seamlessly. Each robot efficiently covers its designated area, minimizing redundancy and gaps in coverage. Additionally, the simulation presents an alternative scenario where the failed robot is replaced by a new one. In this case, the path of the newly added robot is illustrated by white lines, demonstrating its navigation to assume the coverage responsibilities of the failed robot.

The proposed methodology demonstrates the systematic exploration and coverage of the entire workspace, ensuring an equitable distribution of workspace areas among the robots without any coverage gaps or overlaps. This achievement underscores the effectiveness of the proposed algorithms in facilitating comprehensive coverage tasks. Importantly, the performance of the algorithm remains consistent regardless of the number of robots in the system, owing to its distributed nature. This scalability aspect is crucial for real-world deployment scenarios, where the number of robots may vary based on task requirements or environmental complexity. Overall, the proposed methodology highlights the robustness and reliability of the approach in achieving efficient coverage in multi-robot systems. In addition to the promising results, this paper introduces several novel contributions in the domain of multi-robot systems and coverage tasks. It introduces a dynamic partitioning approach using the Manhattan Voronoi method to efficiently allocate coverage areas to robots based on their initial positions, thereby ensuring non-overlapping coverage. The inclusion of fault tolerance mechanisms, such as Voronoi cell reallocation and replacement of failed robots, enhances the adaptability of the system to recover seamlessly from robot failures. Moreover, the proposed methodology emphasizes efficient resource utilization, including energy conservation and minimizing redundant movements, addressing practical concerns in real-world deployment scenarios. The validation of proposed methodologies in the V-rep simulation environment bolsters the novelty by offering a realistic platform for testing and evaluating multi-robot systems. Collectively, the integration of innovative algorithms, fault tolerance mechanisms, resource efficiency strategies, and validation in a realistic simulation environment contributes to the novelty of the proposed methodology in advancing the field of multi-robot systems and coverage tasks.

VII. CONCLUSION

This paper introduces a methodology designed to address robot failures within multi-robot systems engaged in coverage path planning using the MRSimExCoverage strategy. The initial partitioning of the workspace is established through a Manhattan distance-based Voronoi partitioning approach, assuming an obstacle-free environment. As robots undertake both coverage and exploration tasks simultaneously, the possibility of robot failures is considered. In response, the paper presents two key strategies: reallocation of Voronoi cell boundaries to ensure contiguous Voronoi cells and the replacement of failed robots. A notable outcome of this methodology is the achievement of a uniform online workspace allocation scheme for multi-robot systems, promoting equitable task distribution among the robots. The simulations were conducted in V-Rep simulation environment, employing the DR12 robot model. Importantly, the robots effectively accomplish coverage and exploration tasks, eliminating coverage gaps or overlaps. In summary, this methodology provides an effective solution for handling robot failures within multi-robot systems engaged in coverage path planning. It ensures efficient workspace allocation, maintains partitioning integrity, and guarantees comprehensive coverage in the presence of robot failures.

In considering future developments for the methodology introduced in this paper, several promising directions emerge. First, extending the methodology to handle dynamic obstacles in the workspace would significantly enhance its applicability in real-world scenarios, ensuring continuous coverage despite environmental changes. Additionally, exploring adaptive partitioning techniques that dynamically adjust Voronoi cell boundaries based on evolving environmental conditions or task requirements could further optimize workspace allocation and improve coverage efficiency. Integrating learning techniques to enable robots to adaptively improve their coverage and exploration strategies over time could lead to more adaptive and efficient multi-robot systems. Finally the implementation of the algorithm in real Multi robot systems also provide a challenging problem. Addressing these future research directions promises to advance the state-of-the-art in multi-robot systems for coverage path planning, leading to more robust, efficient, and adaptable solutions for various applications.

AUTHOR CONTRIBUTIONS

Vishnu G. Nair: Idea generation, software simulation, writing and submitting. **M. V. Dileep:** Writing, simulation, formatting and editing. **K. R. Guruprasad:** Overall supervision and review.

DECLARATIONS

FUNDING AND CONFLICTS OF INTERESTS/COMPETING INTERESTS

The authors declare that no competing financial interest or personal relationship could have appeared to influence the work reported in this paper. The authors have no relevant financial or non-financial interests to disclose.

AVAILABILITY OF DATA AND MATERIALS

The data supporting this study's findings are available from the corresponding author upon reasonable request.

ACKNOWLEDGMENT

The authors thank Manipal Academy of Higher Education, Manipal, for providing the facilities needed for the proposed research work.

REFERENCES

[1] X. Liu, Z.-R. Peng, L.-Y. Zhang, and Q. Chen, "Real-time and coordinated UAV path planning for road traffic surveillance: A penalty-based boundary intersection approach," *Int. J. Control, Autom. Syst.*, vol. 20, no. 8, pp. 2655–2668, Aug. 2022.

[2] J. Yu, Z. Chen, Z. Zhao, X. Wang, Y. Bai, J. Wu, and J. Xu, "Smooth path planning method for unmanned surface vessels considering environmental disturbance," *Int. J. Control, Autom. Syst.*, vol. 21, no. 10, pp. 3285–3298, Oct. 2023.

[3] X. Lv, W. Li, and J. Wang, "Safety-field-based path planning algorithm of lane changing for autonomous vehicles," *Int. J. Control, Autom. Syst.*, vol. 20, no. 2, pp. 564–576, Feb. 2022.

[4] G. Wang, J. Xiao, R. Xue, and Y. Yuan, "A multi-group multi-agent system based on reinforcement learning and flocking," *Int. J. Control, Autom. Syst.*, vol. 20, no. 7, pp. 2364–2378, Jul. 2022.

[5] J. Kim, D. Jang, and H. J. Kim, "Distributed multi-agent target search and tracking with Gaussian process and reinforcement learning," *Int. J. Control, Autom. Syst.*, vol. 21, no. 9, pp. 3057–3067, Sep. 2023.

[6] A. Janchiv, D. Batsaikhan, B. Kim, W. G. Lee, and S.-G. Lee, "Time-efficient and complete coverage path planning based on flow networks for multi-robots," *Int. J. Control, Autom. Syst.*, vol. 11, no. 2, pp. 369–376, Apr. 2013.

[7] C. Hu, Z. Meng, G. Qu, H.-S. Shin, and A. Tsourdos, "Distributed cooperative path planning for tracking ground moving target by multiple fixed-wing UAVs via DMPC-GVD in urban environment," *Int. J. Control, Autom. Syst.*, vol. 19, no. 2, pp. 823–836, Feb. 2021.

[8] H. Choset, "Coverage of known spaces: The boustrophedon cellular decomposition," *Auto. Robots*, vol. 9, pp. 247–253, Jun. 2000.

[9] Y. Gabriely and E. Rimon, "Competitive on-line coverage of grid environments by a mobile robot," *Comput. Geometry*, vol. 24, no. 3, pp. 197–224, Apr. 2003.

[10] T. W. Min and H. K. Yin, "A decentralized approach for cooperative sweeping by multiple mobile robots," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. Innov. Theory, Pract. Appl.*, Aug. 1998, pp. 380–385.

[11] S. Hert and V. Lumelsky, "Polygon area decomposition for multiple-robot workspace division," *Int. J. Comput. Geometry Appl.*, vol. 8, no. 4, pp. 437–466, Aug. 1998.

[12] M. Jager and B. Nebel, "Dynamic decentralized area partitioning for cooperating cleaning robots," in *Proc. IEEE Int. Conf. Robot. Autom.*, Aug. 2002, pp. 3577–3582.

[13] I. Rekleitis, A. P. New, E. S. Rankin, and H. Choset, "Efficient boustrophedon multi-robot coverage: An algorithmic approach," *Ann. Math. Artif. Intell.*, vol. 52, nos. 2–4, pp. 109–142, Apr. 2008.

[14] K. R. Guruprasad, Z. Wilson, and P. Dasgupta, "Complete coverage of an initially unknown environment by multiple robots using Voronoi partition," in *Proc. 2nd Int. Conf. Adv. Control Optim. Dyn. Syst.*, 2012, pp. 1–19.

[15] V. G. Nair and K. R. Guruprasad, "Manhattan distance based Voronoi partitioning for efficient multi-robot coverage," in *Control Instrumentation Systems*. Cham, Switzerland: Springer, 2020, pp. 81–90.

[16] K. R. Guruprasad and P. Dasgupta, "Distributed spatial partitioning of an initially unknown region for a multi-robot coverage application," in *Proc. 13th Int. Conf. Auto. Agents Multiagent Syst.*, 2012, pp. 1453–1454.

[17] K. Hungerford, P. Dasgupta, and K. R. Guruprasad, "A repartitioning algorithm to guarantee complete, non-overlapping planar coverage with multiple robots," in *Tracts in Advanced Robotics*. Cham, Switzerland: Springer, 2016, pp. 33–48.

[18] V. G. Nair and K. R. Guruprasad, "GM-VPC: An algorithm for multi-robot coverage of known spaces using generalized Voronoi partition," *Robotica*, vol. 38, no. 5, pp. 845–860, May 2020.

[19] V. G. Nair and K. R. Guruprasad, "Centroidal Voronoi partitioning using virtual nodes for multi robot coverage," *Int. J. Eng. Technol.*, vol. 7, no. 2, pp. 135–139, 2018.

[20] V. G. Nair and K. R. Guruprasad, "MR-SimExCoverage: Multi-robot simultaneous exploration and coverage," *Comput. Electr. Eng.*, vol. 85, Jul. 2020, Art. no. 106680.

[21] V. G. Nair and K. R. Guruprasad, "Geodesic-VPC: Spatial partitioning for multi-robot coverage problem," *Int. J. Robot. Autom.*, vol. 35, no. 3, pp. 189–198, 2020.

[22] V. G. Nair and K. R. Guruprasad, "2D-VPC: An efficient coverage algorithm for multiple autonomous vehicles," *Int. J. Control, Autom. Syst.*, vol. 19, no. 8, pp. 2891–2901, Aug. 2021.

[23] V. G. Nair, R. S. Adarsh, K. P. Jayalakshmi, M. V. Dileep, and K. R. Guruprasad, "Cooperative online workspace allocation in the presence of obstacles for multi-robot simultaneous exploration and coverage path planning problem," *Int. J. Control, Autom. Syst.*, vol. 21, no. 7, pp. 2338–2349, Jul. 2023.

[24] K. Hungerford, P. Dasgupta, and K. R. Guruprasad, "Distributed, complete, multi-robot coverage of initially unknown environments using repartitioning," in *Proc. Int. Conf. Auto. Agents Multi-Agent Syst.*, 2014, pp. 1453–1454.

[25] A. Gautam, S. P. Arjun Ram, V. S. Shekhawat, and S. Mohan, "Balanced partitioning of workspace for efficient multi-robot coordination," in *Proc. IEEE Int. Conf. Robot. Biomimetics*, Dec. 2017, pp. 104–109.

[26] A. Gautam, V. S. Shekhawat, and S. Mohan, "A graph partitioning approach for fast exploration with multi-robot coordination," in *Proc. IEEE Int. Conf. Syst. Man Cybern.*, Oct. 2019, pp. 459–465.

[27] A. Gautam, B. Jha, G. Kumar, J. K. Murthy, S. A. Ram, and S. Mohan, "FAST: Synchronous frontier allocation for scalable online multi-robot terrain coverage," *J. Intell. Robotic Syst.*, vol. 87, nos. 3–4, pp. 545–564, Sep. 2017.

VISHNU G. NAIR received the B.Tech. degree from Kerala University, India, in 2010, the M.Tech. degree from Manipal Institute of Technology, Manipal, India, in 2012, and the Ph.D. degree from the National Institute of Technology Karnataka, India, in 2019. He is currently with the Department of Aeronautical Engineering, Manipal Institute of Technology, Manipal Academy of Higher Education. His research interests include control, motion planning, and multi-robotic systems.

M. V. DILEEP received the B.Tech. degree from Kerala University, India, in 2010, the M.Tech. degree from Manipal Institute of Technology, India, in 2012, and the Ph.D. degree from Manipal University, Manipal, India, in 2015. He was with the Department of Aerospace Engineering, Chungnam National University, South Korea. His research interests include control, trajectory optimization, and UAVs.

K. R. GURUPRASAD received the B.E. degree from Mangalore University, India, and the M.Sc. and Ph.D. degrees from Indian Institute of Science, Bengaluru, India, in 1997 and 2009, respectively. He is currently with the Department of Mechanical Engineering, Indian Institute of Technology Kanpur, India. His research interests include control, motion planning, and multi-robotic systems and applications.