Defence Science Journal, Vol. 75, No. 5, September 2025, pp. 546-555, DOI : 10.14429/dsj.20776

Online Exploration Path Planning for a Ground Robot Using the Structure of the Environment

Nitin Kumar Dhiman[#,*], Aniket Kalra[#], K. Dali Naidu[#], and Deepak Khemani[!]

[#]DRDO-Centre for AI & Robotics (CAIR), Bengaluru - 560 093, India
[!]Plaksha University, Mohali - 140 306, India
[*]Email: nitinkdhiman@gmail.com

ABSTRACT

In deploying robots for security and safety applications, the robot should fully explore one region of the environment, such as a room, before proceeding to other regions. To achieve such a behaviour, this article proposes an enhancement of the hierarchical exploration algorithm titled 'Technology for Autonomous Robotics Exploration' (TARE). The algorithm uses region predictions to select appropriate parameters, with regions being user inputs or algorithm outputs. The enhancement includes a new viewpoint selection algorithm which uses a region's boundary to filter the viewpoint choices for local path formation, ensuring full exploration of the region before moving elsewhere. The algorithm is assessed by two criteria: (1) active viewpoints' count during exploration, and (2) cumulative wait time for viewpoints visit *i.e.*, from identification of a viewpoint to robot's visit to the viewpoint. Simulations were conducted in Long-T and Garage environments. In Long-T and Garage environments, the proposed algorithm's maximum active viewpoints were 10 and 12, versus TARE's 37 and 60. The average time to visit a viewpoint is 43 and 289 sec. for the proposed algorithm vs 294 and 448 sec. for TARE's. Future work will evaluate manipulation actions on objects in the environment to aid exploration and coverage.

Keywords: Exploration; Coverage path planning; Active perception; Simultaneous localization and mapping (SLAM)

1. INTRODUCTION

Search and rescue, area coverage, and inspection of an unknown environment are among the common applications of autonomous robots[1,2]. These applications make exploration and coverage in an unknown area key capabilities of autonomous robots. Exploration is the capability of a system to identify a goal or a path to move toward the unknown region to gain awareness of the complete environment as soon as possible[3]. Coverage refers to sweeping the area with a sensor so that every point in the environment is within the sensor range during the traversal by the robot[4].

In large-scale 3D environments, challenges include (*i*) efficiently covering the area without oscillations and redundancy in travel paths, (*ii*) managing computational load, and (*iii*) quickly covering identified sub-regions[5-6]. Robots are typically equipped with sensors *e.g.*, 3D LiDAR and optical cameras *etc.*, for navigation. They may also have payload sensors *e.g.*, chemical detectors and specific cameras for object detection *etc.* which generally have shorter ranges.

In this article, the focus of the exploration and coverage solution is on completing coverage of sub-regions of the environment as soon as possible [point (*iii*)]. The navigation sensor's perception range, being greater than a coverage sensor's, is used to predict the environment's structure (region type and polygonal boundaries *etc.*). Consider an operational requirement of exploration and coverage of the system in the identification of an object or chemical presence in an unknown environment. For safety, deciding early on to cover a part of the environment is preferred. Each detected or given part, *e.g.*, a room, corridor, or bounded open region *etc.*, should be fully covered before exploring another area. The TARE[5] algorithm is one of the better-known solutions for online exploration and coverage. An instance of planning run with the TARE exploration algorithm is shown in Fig. 1. The image shows the robot travelling a long corridor but not fully covering the side walls, which may not be desirable for some security and safety applications. This article proposes a solution to prevent such an expected behaviour. The contribution made in this article includes a proposal for enhancement of the hierarchical exploration and coverage planning framework with the following:

- A viewpoint selection algorithm to prioritize the completion of the exploration and the coverage of a region. The region is assumed as input to the system.
- Generate local coverage paths based on the bounded region.

The control flow is implemented using a Behavior-Tree[7] on ROS2[8]. The terrain analysis module and the Garage simulation environment from Autonomous Exploration Development Environment (AEDE)[9] are used. The first claim is that the average wait time for a viewpoint is reduced. The wait time is defined as the time elapsed from selecting a viewpoint to its

Received : 06 February 2025, Revised : 02 March 2025
Accepted : 22 April 2025, Online published : 01 September 2025

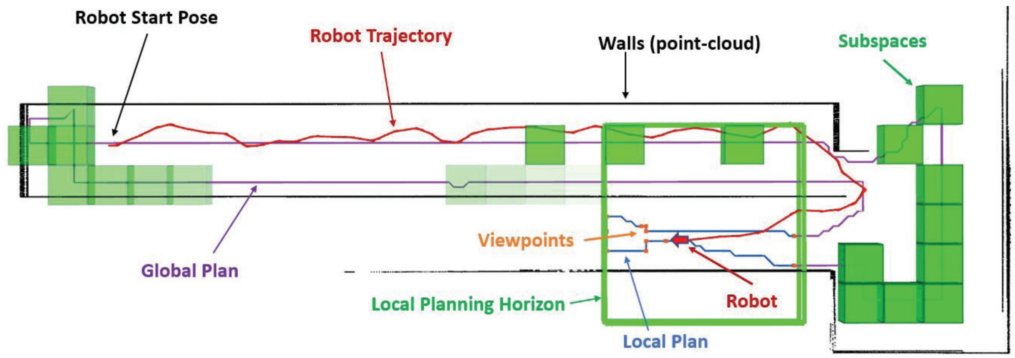

Figure 1. (Best viewed in colour) A snapshot of the exploration process in a simulation world. The coverage sensor range is 12 m. The green square shows the part of the space (called subspace) that contains one or more viewpoint positions for the robot to visit.

being visited by the robot. Since the choice of a viewpoint to visit will increase the wait time for some other viewpoints, the first claim implies that the cumulative sum of wait time for all viewpoints will be reduced. The second claim asserts that the number of viewpoints yet to be visited remains minimal during the exploration.

Section II presents a review of the literature. Section III provides a formal definition of the problem. Section IV explains the proposed algorithm. Section V presents the results of the simulation study.

2. LITERATURE REVIEW

Solutions to autonomous exploration and coverage of an unknown region have been attempted from multiple approaches. A brief discussion of approaches follows:

2.1 Frontier-Based Approaches

A frontier[10] defines a boundary between the known (free and obstacle) and unknown regions on the map. The map is represented as a grid of cells or voxels that are labelled as known or unknown. For environments in 3D, analogous to a frontier is known as a surface point[11]. A robot gains spatial awareness on moving through the frontier, i.e. moving from the known towards the unknown region. The literature focuses on the selection of the frontier among the candidates based on area or volume gain[12-13] energy efficiency[14], change in travel speed and heading[15-16], or a utility function which combines area or volume gain and distance travel as cost components[11,13] *etc*. Information-theoretic approaches of frontier selection aim to reduce the uncertainty in the output of Simultaneous Localization And Mapping (SLAM) algorithms by evaluating the choice of a viewpoint for loop closure as part of the process. Such works are also known as Active-SLAM. A survey on Active-SLAM has been presented by Placed *et. al*.[17]. Most such approaches assume the presence of one sensor for both exploration and coverage. In our work, the coverage of the environment is performed with an additional sensor, the range of which influences the selection of viewpoints.

2.2 Sampling-Based Approaches

A sampling-based approach draws a random sample for the next goal of exploration in the neighbourhood, checks whether a robot can travel to it, and arranges them in a Rapidly-Exploring Random Tree (RRT)[18] or in a Rapidly-exploring Random Graph (RRG)[19] like data structures for selection of the path. The next waypoint of the path is selected based on the maximum information gain[20], speed of travel[21] *etc*. Approaches selecting goals based solely on neighbourhood samples make short-sighted decisions, leading to suboptimal solutions. Some approaches solve this problem by combining the process with global planning; some such works are mentioned next.

2.3 Hierarchical Exploration Approaches

The hierarchical exploration algorithms divide the computation at two levels: local and global. The local planning keeps computation cost within bounds by solving the exploration at fine resolution for a sliding window region around the robot. Global planning is performed at the coarse representation of the whole environment, guiding the direction of local exploration. Such hierarchical approaches can overcome challenges arising because of short-sighted decision horizons.

In GBPlanner[6,16], UFOExplorer[22], FAEL[23] and STAGE[24], the local planner utilizes sampling-based approaches. FAEL uses a utility measure combining movement distance, information gain and coverage to select a path for the robot to follow. STAGE maintains a global traversability graph to manage the dynamic changes in the environment, *e.g.*, the closure of a pathway. In TARE[5,25] as well, the local planning utilizes a high-resolution environment map within a moving window centered on the robot, called the local planning horizon (*LPH*). The viewpoints are selected within *LPH* from a lattice of candidate viewpoints based on volumetric gain. The selected viewpoints are arranged in a travelling salesman problem (TSP) path. The global plan abstracts the environment into a coarser grid. Each cell in a coarser grid is called a subspace. The cells containing the active viewpoints are arranged in the TSP path. This paper presents an extension to the TARE algorithm.

2.4 Partition Area, Semantic Influence Approaches

To localize the computation of exploration paths to a region, many approaches divide the spatial area into multiple regions. A Voronoi diagram has been built from the 2D occupancy map of the environment to extract the rooms[26], doors[27], and layout of the environment[28,29]. This layout can be used for planning the paths. The proposed algorithm also utilizes a similar method for the prediction of regions. The volumetric exploration algorithm[30] has used object semantics for concurrent environmental inspection. In this approach, the planner operates for a set time, selecting viewpoints to fill the unknown part (holes) of the map. Once holes are filled, it plans the inspection with a coverage sensor[30]. In our algorithm, the exploration and coverage are performed concurrently rather than in a time-interleave manner. In HPHS[31,32], the SLAM map is divided into uniform-size sub-regions. As the environment grows, so does the size of sub-regions. Exploration follows a global plan dictating their order based on a revenue function encompassing travel distance, orientation, and information gains. The proposed algorithm also uses regions for bounding the viewpoint selection area but the region is predicted based on the structure of the environment rather than on uniform size region decomposition of the space.

2.5 Online Coverage Path Planning (CPP) Approaches

The papers[33,34] present a detailed survey of CPP algorithms. Online CPP approaches perform the decomposition of area into multiple regions/cells which are traversed in an appropriate pattern or order for a coverage task. The cell size could be fixed at the start or approximated and varied at run time[35] based on the coverage tool size or robot footprint. A Turing machine for exploration on a two-dimensional multi-level tape has also been used[36]. The tape stores the environment as a hierarchical multi-scale tile, one scale at one level. An approximate cellular decomposition-based representation of the environment enables completeness of coverage, even when a part of the cell is occupied by an obstacle[37]. An approach[38] abstracts the environment into convex shape sectors which are connected by a Hamiltonian path, providing a visitation order of sectors. A minimum set of viewpoints is sampled along the path, connected by a TSP path, for performing the maximum coverage. The challenge with such approaches is the approximation of the cell size based on the coverage sensor tool, which could lead to uncovered spaces in the presence of obstacles if the cell size is too big.

In many security applications, it is expected that a part of the environment can be covered and cleared for other activities by the operator. The focus of most of the above-discussed approaches is to explore and cover the environment as soon as possible without the specific intention of avoiding long wait times for identified viewpoints or covering an identified part of the environment before moving to other parts of the environment. This article presents an extension to TARE which utilizes the awareness of the environment in a selection of waypoints to solve this problem. The viewpoint selection in *LPH* is computed such that the region is fully covered before moving out of it. An area is fully explored when no further unseen area remains. A longer-range navigation sensor is utilized for region prediction, while the coverage sensor is employed for viewpoint selection for coverage of the region. The local planning is performed at a fine resolution so that uneven surfaces and parts of the environment can be covered.

3. PROBLEM DEFINITION

Let's assume that the robot is equipped with a 3D LiDAR-like navigation sensor S_{nav} with range R_{nav} and 360° field of view. The robot also has a payload coverage sensor S_{cov} with range R_{cov} and 360° field of view. $R_{cov} << R_{nav}$. The following assumptions have been made about the sensors: (*a*) the sensors S_{nav} and S_{cov} do not degrade with the change in environmental conditions during the exploration mission, (*b*) the coverage sensor provides certain detection within its perception range, (*c*) the sensors are co-located such that they do not interfere with each other. The deployment region under consideration is a bounded volume $V \in R^3$, which consists of many surfaces.

The volume V consists of free regions (V_{free}) and occupied regions (V_{occ}). A viewpoint $v \in \mathbb{SE}(3)$ defines the pose of the sensor onboard the robot so that awareness and coverage of an unknown environment are obtained. $L^t_{nav} \in \mathbb{SE}(3)$ defines a set of selected viewpoints by the robot till time t. $L^t_{visited} \subset L^t_{nav}$ defines a set of viewpoints visited by the robot till time t or which are no longer providing awareness about unknown space. The set of active viewpoints, L^t_{active}, consists of viewpoints which have not been visited by the robot *i.e.*, $L^t_{active} = L^t_{nav} - L^t_{visited}$. A viewpoint is active till it is not visited by the robot or the area to be seen from the viewpoint has been seen by the robot from some other location. Since the sensors' placement is fixed *w.r.t.* the robot-base frame, the location of the sensor viewpoints is derived from the robot's trajectory. $F_v \in V$ be the surfaces perceived by S_{cov} sensor at a viewpoint v. The perceived surfaces till time t are defined as

$$V_L = \cup_{v \in L^t_{visited}} F_v \tag{1}$$

$\bar{V}=V-V_L$ denotes the not-covered surfaces. Let's consider that V consists of mutually exclusive n regions $B = \{B_1, B_2, B_3, ..., B_n\}$. V = $\{B_1 \cup B_2 \cup B_3 \cup ... \cup B_n\}$. These regions can be detected in the point-cloud map created using S_{nav}. For a viewpoint $v_i \in L^t_{nav}$, q^i_{start} and q^i_{finish} define the time of selection of a viewpoint and the time till the viewpoint is active, respectively.

The algorithm aims to generate a volumetric map of the environment using S_{nav} *i.e.*, all surfaces are mapped, each region is seen using the coverage sensor S_{cov} and coverage is completed as soon as possible. A voxel is said to be visited if it is within the range (R_{cov}) from any point in the robot's trajectory.

Problem: Given $\bar{V}$, B, R_{nav}, the current pose of the robot p_{robot}, the set of active viewpoints L^t_{active}, find a path $P = \{v_1, v_2, ..., v_n\}$ for the robot, when followed, leading to the observation of surfaces in V such that the cumulative wait time spent covering the viewpoints (denoted as Q) is reduced as defined in Eqn. (3) and the count (denoted as A) of active viewpoints *i.e.*, size of L^t_{active} at any time $t \in T$, is minimal, where T is the time instance at which the exploration finishes. A is defined in Eqn. (5).

$$Q_i = q^i_{finish} - q^i_{start}, \forall v_i \in L^t_{active} \tag{2}$$

$$Q = min \sum_{i=1}^{n} Q_i, \ \forall v_i \in L^t_{nav} \tag{3}$$

$$Q_{avg} = \frac{Q}{|L^t_{nav}|} \tag{4}$$

$$A = min\{max\{|L^t_{active}| \ |\forall t \in \{1,2,...,T\}\}\} \tag{5}$$

Eqn. (3) and Eqn. (5) define the first and second claims, respectively. The algorithm which results in lower values for Q_{avg}, Q and A provides better results.

4. METHODOLOGY

The methodology employs hierarchical exploration, where local planning computes viewpoints and coverage within a sliding window around the robot, and global planning retains non-visited viewpoints as a path to guide the local horizon.

Algorithm 1 Hierarchical Exploration Algorithm

Input: p_{robot}, LPH, $min\_count\_unknown$, SLAM point-cloud map M_{pc}, map (a 3D array) for coverage memory $M_{coverage}$
Output: path for robot P, $M_{coverage}$

Initialize $last\_region\_class = open\_space$
Compute $region\_class$, region perimeter boundary as polygon B, $region\_entry\_point$ using the Section IV-C
Initialize sets, $L_B \leftarrow \{\}, L_{LPH-B} \leftarrow \{\}$
loop
  if $|L_B| = 0 \;\; \& \;\; |L_{LPH-B}| > 0$ **then**
    Find a viewpoint $v \in L_{LPH-B}$, nearest to p_{robot}
    $P = ComputeShortestPath(p_{robot}, v)$
    $SendPathToRobot(P)$
    After the robot reaches v, initiate one recursive iteration of the Hierarchical Exploration Algorithm with p_{robot}, $M_{coverage}$, and M_{pc}
  else
    $UpdateCoverageMap(M_{coverage})$
    if p_{robot} is not within B **then**
      if $region\_entry\_point$ not available **then**
        $region\_entry\_point \leftarrow p_{near}$, nearest point to the robot inside B
      end if
      $P_1 \leftarrow ComputeShortestPath(p_{robot}, region\_entry\_point)$
      $SendPathToRobot(P_1)$
    end if
    if $last\_region\_class \neq region\_class$ **then**
      Select parameters, LPH and $min\_count\_unknown$ based on $region\_class$
      Update the algorithm's parameters
    end if
    compute viewpoints using $region\_polygon$ using Algorithm 2, store result in L_B and L_{LPH-B}
    $P_{local} = Compute\_local\_path()$ using Algorithm 3
    compute global path P_{global}
    $last\_region\_class \leftarrow region\_class$
    $SendPathToRobot(P_{local})$
    $UpdateCoverageMap(M_{coverage})$
  end if
end loop

The local exploration planning is enhanced with predictions of the environmental structure, identifying if the robot is in an open area or a closed area *e.g.*, such as a room or corridor etc. This approach improves solutions, as defined in Section 3. It is assumed a module can predict attributes and boundaries of the spatial region in which the robot is currently standing. The region's class is used to select algorithm parameters like the Local Planning Horizon (*LPH*) and the minimum count of frontier/surface cells for considering a cell cluster as a frontier or surface (*min_count_unknown*) *etc*. As the robot explores, the algorithm identifies new viewpoints, within *B*, and outside *B* (but within *LPH*), denoted as, sets L_B and L_{LPH-B} respectively. Once the robot covers *B*, it selects a viewpoint from L_{LPH-B} for further exploration. The pseudo-code is detailed in Algorithm 1. The system uses a timer-based ticking mechanism in the behaviour tree to ensure the exploration of the current region before processing the next. Figure 2 illustrates the system diagram. The modules are discussed in subsequent sections.

4.1 Simultaneous Localization and Mapping (SLAM)

SLAM module provides the state-estimation *i.e.*, current robot pose and aggregated map of the environment. Since the navigation sensor is a range sensor, an online SLAM algorithm with input as point-cloud data is suitable for integration. It is assumed that the SLAM module will provide consistent robot localization and registered point clouds. In our experimentation, registered scans from the simulators have been used.

4.2 Navigation Modules

ROS2 is used as a framework for the development and integration of the modules. The modules interact using ROS2's service and action interface for control and data flow. Algorithm 1 implementation is based on Behavior-Tree[7]. A behaviour tree enables ease of composition of behaviour based on different conditions without changes to exploration's core source code.

4.3 Computation of Semantic Class of the Region

A region class detection module is implemented based on earlier works[27-29], which processes point-cloud data and

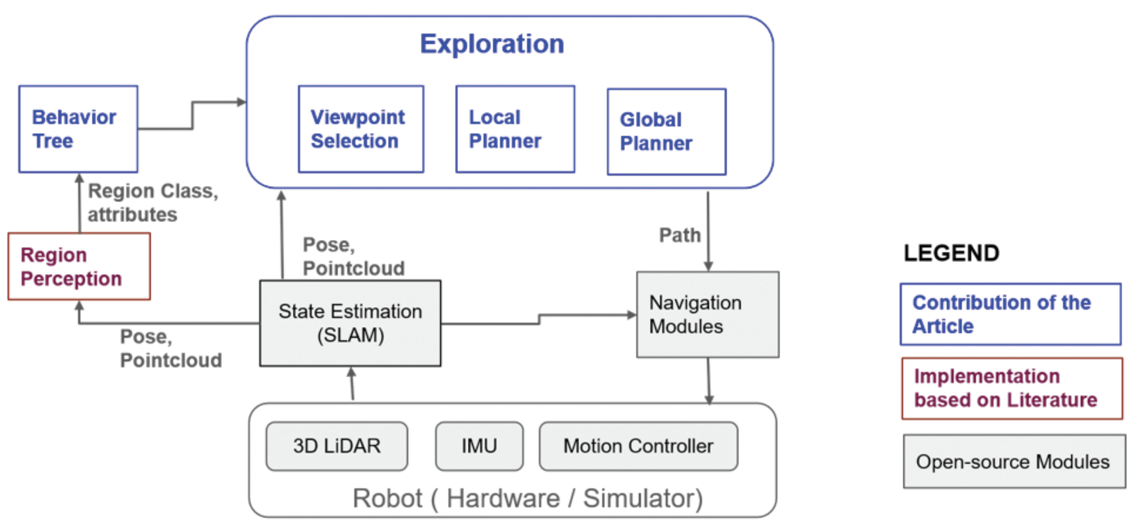

Figure 2. High-level diagram of the exploration system. The exploration modules are built over the open-source TARE5 source.

map data generated using the navigation sensor (3D LiDAR). Online inputs from the operator could also provide the region and its attributes as input to the algorithm. Identification of the semantic class of a region and navigation through the environment can also be implemented using CNN-based solutions[40]. Since detection of the region and its attributes is not the focus of this work, such solutions have not been explored in this article.

Algorithm 2 Viewpoint selection algorithm

Input: $p_{robot}, B, LPH, M_{pc}, M_{coverage}, min\_count\_unknown$, viewpoint sampling resolution r, R_{cov}
Output: L_B, L_{LPH-B}

$F \leftarrow \{\}$, initialize set of frontier/surface cell as empty.
Compute $M_{segment}$ as part of the map M_{pc} covering the LPH and additional range R_{cov} meters around LPH in X-Y plane.
$F \leftarrow ComputeFrontier\&SurfaceCells(M_{segment})$
$\forall f \in F, Status[f] = false;$
$L_{vp\_candidate} = GetViewPointCandidates(LPH, r)$
for $\forall v \in L_{vp\_candidate}$ **do**
  $Visibility[v] = \{\forall f \in F | Status[f] = false, Distance(v, f) < R_{cov}, LineOfSight(v, f) = true\}$
end for
Arrange $v \in L_{vp\_candidate}$ as priority queue PQ based on $|Visibility[v]|$
Initialize set $S_B \leftarrow \{\}$ and $S_{LPH-B} \leftarrow \{\}$
$v_{top} \leftarrow pop(PQ)$
while $|Visibility[v_{top}]| >= min\_count\_unknown$ **do**
  if v_{top} is located inside B **then**
    $L_B \leftarrow L_B \cup v_{top}$
    for $f \in Visibility[v_{top}]$ **do**
      if f is located inside B **then**
        $Status[f] \leftarrow true$
      end if
    end for
  else
    $L_{LPH-B} \leftarrow L_{LPH-B} \cup v_{top}$
    for $f \in Visibility[v_{top}]$ **do**
      if f is not located inside B **then**
        $Status[f] \leftarrow true$
      end if
    end for
  end if
  $L_{vp\_candidate} \leftarrow L_{vp\_candidate} - v_{top}$. Update $Visibility[v]$ based on $Status[f]$.
  Update priority queue PQ based on updated $|Visibility[v]|$.
  if PQ is not empty **then**
    $v_{top} \leftarrow pop(PQ)$
  elsebreak
  end if
end while

4.4 Viewpoint Selection

A viewpoint defines a pose (3D) of the sensor on the robot. If a robot visits a viewpoint, it will lead to an unfolding of unknown space in the environment. In this work, a set of viewpoints is selected to view both the frontiers and surface cells. This approach allows exploration and coverage of the large empty region *i.e.*, which lacks the presence of 3D surfaces in the environment. As the robot travels to a viewpoint, the area in line-of-sight of the S_{cov} sensor is marked as covered. The pseudo-code for the algorithm is shown in Algorithm 2.

Similar to TARE, the viewpoints are sampled from a uniform pattern lattice in local planning horizon *LPH*, and the predicted region (*B*), if available. A minimum set of viewpoints, L_B, is selected to cover *B*. If *LPH* is bigger than *B*, then viewpoints (from L_{LPH-B}) are selected to cover the space of *LPH*. For each viewpoint, an information gain is calculated as a count of frontier cells and surface cells as visible from the coverage sensor (S_{cov}). Due to sub-modularity in exploration[41], the overlapping fields of view among multiple viewpoints can reduce the potential information gain of viewpoints on selecting a viewpoint, which necessitates an adjustment of information gain for remaining viewpoints. Information gain for a viewpoint *v* within *B* is computed considering only the area inside *B*. Figure 3 shows viewpoint v_{i2} seeing cells C_2 and C_3 outside the boundary, which are excluded from its gain calculation. Frontier and unknown cells like C_2 are seen by viewpoints (*e.g.*, v_{o1}) beyond the boundary. External viewpoints within the *LPH* enable coverage of other regions, consequent to the coverage of *B*. Without this, exploration might wrongly assume no new regions exist beyond the current input.

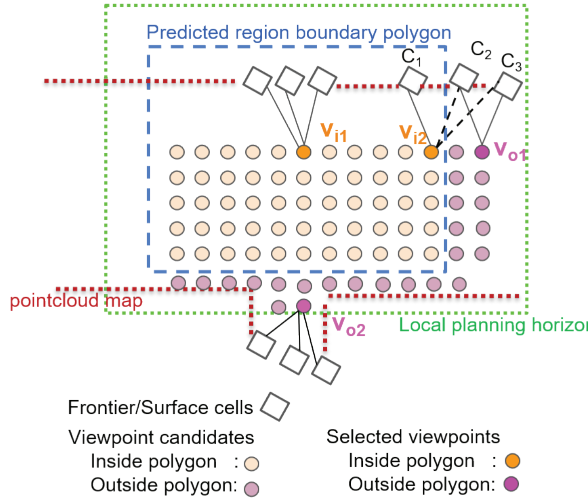

Figure 3. The figure shows the visibility of frontier cells from viewpoints. The dashed edges show the visibility of cells outside the predicted region boundary by viewpoint v_{i2}.

Algorithm 3 Computing_local_path

▷ Compute a feasible path for the robot to trace
Input: p_{robot}, region B, view-points L_B which are inside the B,
Output: Local path P_{local}

for $v_i, v_j \in L_B \cup p_{robot}$ **do**
  computes the shortest path and distance,
  store path at location (i, j) in path matrix PM
  store distance at location (i, j) in distance matrix DM
end for
Compute a TSP sequence Seq using matrix DM and p_{robot} as the start position;
Store viewpoints in order of Seq in P_{temp}.
for every viewpoint pair $v_i, v_{i+1} \in P_{temp}$ **do**
  retrieve path between viewpoints v_i and v_{i+1} as $PM_{i,i+1}$ from PM.
  Smoothen the path $PM_{i,i+1}$ [5]
  $P_{local} \leftarrow P_{local} + PM_{i,i+1}$
end for

4.5 Local Planning: Forming a Path with Viewpoints

The algorithm for composing a path with the selected viewpoints is listed in Algorithm 3, which is as in TARE. The

Table 1. Results of simulation experiments

World name	Method	Coverage (%)	Travel distance (m)	Exploration time(sec)	Q_{avg} (sec)	Q (sec)	A
Long-T	TARE	100	1083±60	647±106	294	11785	37
	Proposed	100	933±120	696 ±105	**43**	**1183**	**10**
	HPHS	34	-	-	-	-	-
Garage	TARE	100	4840±260	2400±143	448	62282	60
	Proposed	100	5200±400	2700±247	**289**	**33637**	**12**
	HPHS	13	-	-	-	-	-

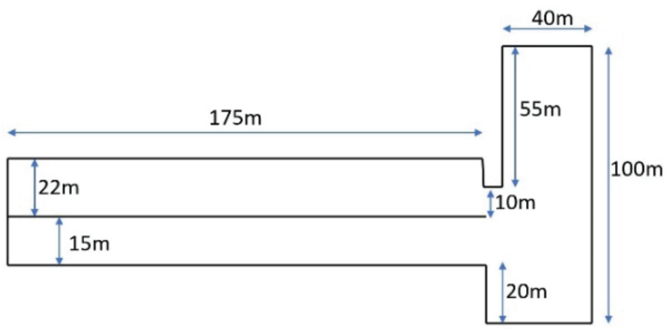

Figure 4. Top-down view of the long-T simulation world.

only exception is that the local planning is restricted to the viewpoints which are inside the region boundary. If an input region boundary polygon is available, it is used to select the viewpoints. These viewpoints are arranged in a TSP path for the robot to follow.

4.6 Global Planning

The global planning process is as in TARE. The main aim of global planning is to guide the direction of exploration when there are no viewpoints available for selection in LPH. For global planning, the physical space is divided into pre-defined size cuboid cells, which are coarser in resolution than those used for local planning. This choice provides computational efficiency advantages.

Each cuboid is known as a subspace[5]. Each subspace may contain a list of selected viewpoints. The subspace could be in either of the following three states: “explored”, “unknown” and “exploring”. If the robot has never visited or covered any part of the subspace, it is in an “unknown” state. If a subspace has viewpoints, it is in an “exploring” state. If the robot has visited all viewpoints i.e., completely seen the space using the S_{cov} sensor, it is marked in the “explored” state. As the robot moves, the unseen surface could become visible, resulting in a change of state from “unknown” to “exploring” to “explored”. The global path connects subspaces which are in an “exploring” state using a TSP path. The connection between the global path and the local path is maintained as in TARE.

5. SIMULATION EXPERIMENT

Simulation experiments were conducted to evaluate the algorithm in travel distance, exploration time, average wait time Q_{avg}, cumulative wait time for viewpoints Q and maximum count of active viewpoints during the exploration (A). The proposed algorithm was compared with the TARE[5,42] and HPHS[31,32]. HPHS also employs the region-based approach for the completion of the exploration; therefore, it is a suitable candidate for comparison. The TARE algorithm is used as the benchmark algorithm as the proposed solution extends this approach.

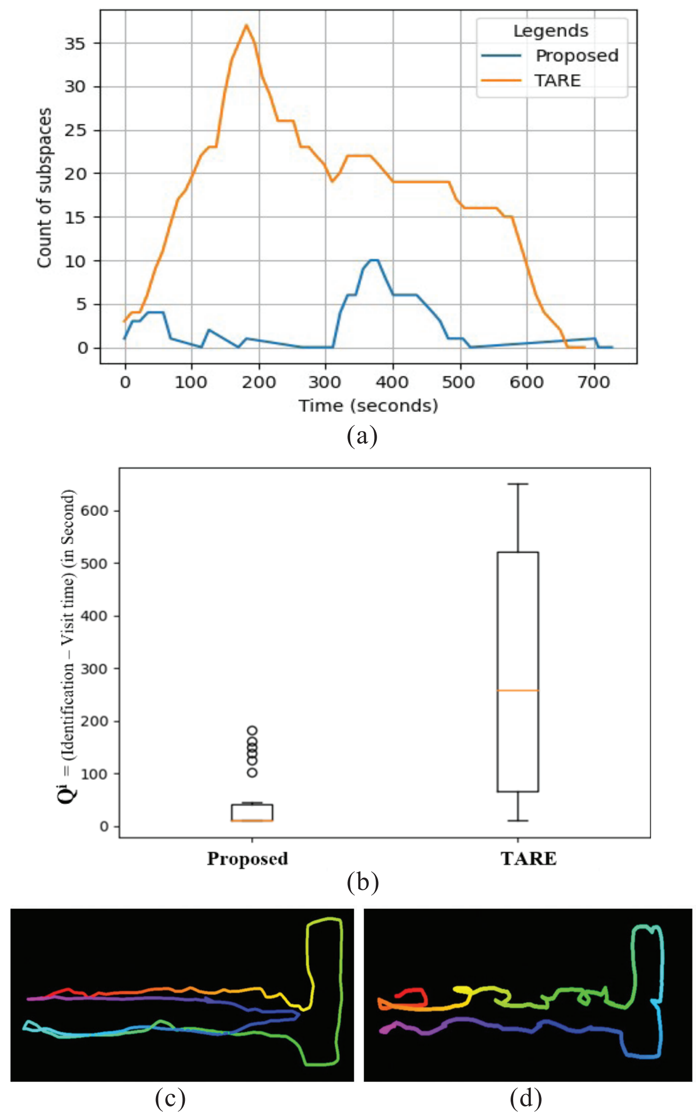

Figure 5. For the Long-T simulation world:(a) Shows the count of active subspaces during the exploration; (b) Statistics for the time spent in clearing a subspace; (c) and (d) Shows the trajectory travelled by TARE and the proposed algorithm. The VIBGYOR colour coding has been used. The last robot pose is shown in violet colour and the start robot pose is in red.

For TARE, the results have been computed for subspaces that are in the "exploring" state. A subspace represents a small volume of space and may contain one or more viewpoints that are planned for a visit by a robot in the future. The viewpoints contained inside a subspace are geometrically near, and the centroid of the subspace can be considered representative of the viewpoints in the subspace. Therefore, a subspace is a sufficient replacement for a viewpoint for comparison of the algorithm.

The simulation test environment and script for the generation of simulation results from AEDE[9] have been adapted and used. The robot is equipped with 3D LiDAR with a 100m range. For the coverage task, data from the same 3D LiDAR sensor acts as a mock-up coverage sensor, with a detection range truncated to 12m. The skid-steer robot is set to travel at a top speed of 2m/sec. The cut-off time for the simulation experiments was set at 60 minutes or 95 per cent of the area covered by the environment. HPHS does not have provision for parametrization of the coverage sensor's range, the 3D LiDAR range of 100 m is used for the mapping module. The presented results are the average of ten runs. Simulation experiments have been performed on two simulation worlds: Long-T and Garage[9].

Figure 4 shows the images of the Long-T simulation world which consists of two long corridors connected to a large open room. The Garage simulation world consists of a five-floor structure of size 140m x 110m. The size of the environment is much bigger than the sensor horizon of S_{cov} and S_{nav}, which makes the task challenging.

Table I lists the results of simulation experiments for both simulation worlds. HPHS algorithm could not complete the exploration as the algorithm was suffering from the following: (*i*) repeated oscillatory assignments of goals before a visit to the existing goal (*ii*) the robot indefinitely trying to reach one goal (*iii*) failure to detect any feasible frontiers. The data from unsuccessful runs of the HPHS has not been considered for analysis.

As shown in Table I, the total travel distance and exploration time for the proposed method are more than those of the TARE algorithm. This additional travel by the robot is attributed to multiple turns by the robot within the *B,* completing the region before moving ahead to other unknown regions. This additional travel is a trade-off for lesser wait time for viewpoints (*i.e.* Q_{avg}, Q). Fig 5(d) shows the trajectory plot of one run of the algorithm on the Long-T simulation world, which shows the turns in trajectory to cover a region. The turns in the robot's trajectory also led to a reduction in travel speed, which resulted in additional travel time.

TARE's exploration path prioritizes the robot's current heading, reducing turns. Because of this behaviour, the algorithm tends to leave some areas partially covered before proceeding to subsequent regions. The algorithm's trajectory requires the robot to return to the start position due to earlier left unvisited viewpoints, which is shown in Fig. 5(c). The count of active subspaces, A, during the exploration, is shown in Fig. 5(a) and Fig. 6(a). This shows that the proposed algorithm is better at maintaining a lesser number of active viewpoints. The box plots in Fig. 5(b) and Fig. 6(b) also show that the proposed

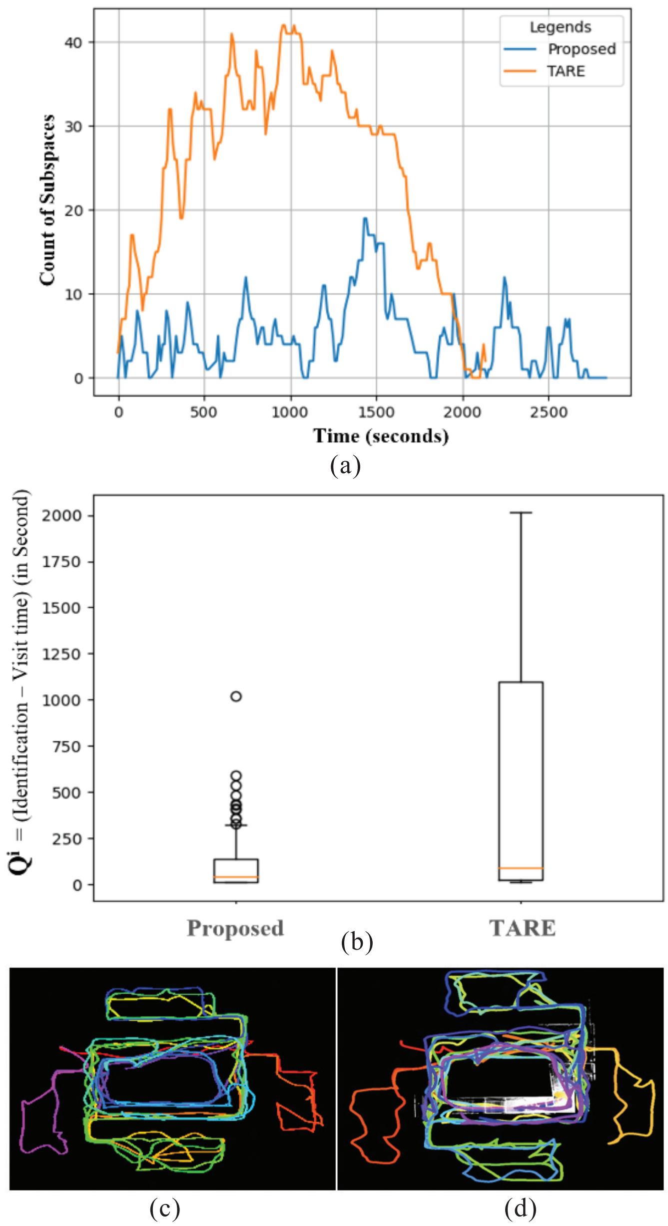

Figure 6. For the Garage world: (a) Shows the count of active subspaces during the exploration; (b) Statistics for the time spent in clearing a subspace; (c) and (d) Shows the trajectory travelled TARE and proposed algorithm.

algorithm results in a shorter wait time for viewpoints. These results demonstrate that the proposed method prioritizes the visit to active viewpoints in the neighbourhood, therefore, covering the region much earlier.

6. CONCLUSION

This article proposes an enhancement of the hierarchical exploration algorithm TARE to satisfy the operational requirement of the security agencies to cover a region before moving to the other regions in the deployment environment. The algorithm uses predictions about the class and boundary estimates of the perceived region to select parameters that influence viewpoint selection and local path formation. A new viewpoint selection algorithm restricts viewpoint choices based on sensor coverage within the predicted region, before selecting viewpoints outside the region for further exploration. Our simulation experiments study demonstrates that the proposed algorithm exhibits the desired behaviour of exploring the current input region before moving to other regions. The proposed algorithm performs better than TARE on the (*i*) average and cumulative wait time of the viewpoints, and (*ii*) count of active viewpoints during the exploration process.

7. FUTURE WORK

One of the limitations of the current work is an assumption about the availability of the prediction of the regions in the environment and perfect sensors' detections throughout the detection range. The proposed algorithm will perform poorly in case of incorrect predictions. Also, the assumption about sensors' accuracy may not hold *e.g.*, for a chemical sensor acting as a coverage sensor the certainty of detection will vary at the detection distance. In case of errors in the navigation sensor, a solution based on the Active-SLAM approach will be required. An enhancement of the existing work will be taken up as future activities to relax these assumptions.

In current work, the robot does not engage with the environment during the exploration and coverage processes, a consideration which may impede effective deployment in cluttered settings. In future, the algorithm will be augmented to assess the manipulation of environmental objects as an action choice, facilitating the creation of more efficient travel paths for improved coverage.

REFERENCES

1. Liu Y, Nejat G. Robotic Urban Search and Rescue: A survey from the control perspective. Journal of Intelligent & Robotic Systems. 2013 Mar 12;72(2):147–65. doi:10.1007/s10846-013-9822-x.
2. Bouman A, Ott J, Kim SK, Chen K, Kochenderfer MJ, Lopez B, Agha-mohammadi AA, Burdick J. Adaptive coverage path planning for efficient exploration of unknown environments. In: Proc IEEE/RSJ Int Conf Intell Robots Syst (IROS); 2022. p. 11916-23. doi:10.1109/IROS47612.2022.9982287.
3. Stachniss C. Robotic mapping and exploration. 55. Springer; 2009
4. Tan CS, Mohd-Mokhtar R, Arshad MR. A comprehensive review of coverage path planning in robotics using classical and heuristic algorithms. IEEE Access. 2021;9:119310-42. doi:10.1109/ACCESS.2021.3108177
5. Cao C, Zhu H, Choset H, Zhang J. TARE: A Hierarchical Framework for Efficiently Exploring Complex 3D Environments. In Robotics: Science and Systems (RSS). 2021 Jul;5. doi:10.15607/RSS.2021.XVII.018
6. Dang T, Tranzatto M, Khattak S, Mascarich F, Alexis K, Hutter M. Graph-based subterranean exploration path planning using aerial and legged robots. J Field Robot. 2020 Dec;37(8):1363-88. doi: https://doi.org/10.1002/rob.21993
7. Colledanchise M, Ögren P. Behavior trees in robotics and AI: An introduction. CRC Press; 2018. doi:10.1201/9780429489105
8. Macenski S, Foote T, Gerkey B, Lalancette C, Woodall W. Robot operating system 2: Design, architecture, and uses in the wild. Sci. Robot. 2022;7(66). doi:10.1126/scirobotics.abm6074
9. Cao C, Zhu H, Yang F, Xia Y, Choset H, Oh J, Zhang J. Autonomous exploration development environment and the planning algorithms. In: 2022 Int Conf Robot Autom (ICRA). p. 8921-8928. doi:10.1109/ICRA46639.2022.9812330
10. Yamauchi B. A frontier-based approach for autonomous exploration. In: Proc IEEE Int Symp Comput Intel Robot Autom (CIRA) 1997. p. 146-151. doi:10.1109/CIRA.1997.613851
11. Senarathne P, Wang D. Towards autonomous 3D exploration using surface frontiers. In: 2016 IEEE Int Symp Safety, Security Rescue Robot (SSRR). 2016. p. 34-41. doi:10.1109/SSRR.2016.7784274.
12. Makarenko AA, Williams SB, Bourgault F, Durrant-Whyte HF. An experiment in integrated exploration. In IEEE/RSJ Int Conf Intel Robot Syst (IROS). 2002. p. 534-9. doi:10.1109/IRDS.2002.1041445.
13. Burgard W, Moors M, Stachniss C, Schneider FE. Coordinated multi-robot exploration. IEEE Transactions on Robotics. 2005;21(3):376-86. doi:10.1109/TRO.2004.839232
14. Mei Y, Lu Y-H, Lee CG, Hu YC. Energy efficient mobile robot exploration. In: 2006 IEEE Inter Conf Robot Autom (ICRA). 2006. p. 505-11. doi:10.1109/ROBOT.2006.1641761
15. Dai A, Papatheodorou S, Funk N, Tzoumanikas D, Leutenegger S. Fast frontier-based information-driven autonomous exploration with an MAV. In: 2020 IEEE Inter Conf Robot Autom (ICRA). 2020. p. 9570-6. doi:10.1109/ICRA40945.2020.9196707.
16. Cieslewski T, Kaufmann E, Scaramuzza D. Rapid exploration with multi-rotors: A frontier selection method for high speed flight. In: 2017 IEEE/RSJ Int Conf Intel Robot Syst (IROS). 2017. p. 2135-42. doi:10.1109/IROS.2017.8206030.
17. Placed JA, Strader J, Carrilo H, Atanasov N, Indelman V, Carlone L. A survey of active simultaneous localization and mapping: State of the art and new frontiers. IEEE Transactions on Robotics. 2023;39(3):1686-705. doi:10.1109/TRO.2023.3248510.
18. LaValle S. Rapidly-exploring random trees: A new tool for path planning. Research Report 9811; 1998.
19. Karaman S, Frazzoli E. Sampling-based algorithms for optimal motion planning. International Journal of Robotics Research. 2011;30(7):846-94. doi:10.1177/0278364911406761.
20. Bircher A, Kamel M, Alexis K, Oleynikova H, Siegwart R. Receding horizon “next-best-view” planner for 3D exploration. In: 2016 IEEE Inter Conf

Robot Autom (ICRA). 2016. p. 1462-8. doi:10.1109/ICRA.2016.7487281

21. Dharmadhikari M, Dang T, Solanka L, Loje J, Nguyen H, Khedekar N, Alexis K. Motion primitives-based path planning for fast and agile exploration using aerial robots. In: 2020 IEEE Inter Conf Robot Autom (ICRA). 2020. p. 179-85. doi:10.1109/ICRA40945.2020.9196964.
22. Duberg D, Jensfelt P. UFOexplorer: Fast and scalable sampling-based exploration with a graph-based planning structure. IEEE Robotics and Automation Letters. 2022;7(2):2487-94. doi:10.1109/LRA.2020.3013861.
23. Huang J, Zhou B, Fan Z, Zhu Y, Jie Y, Li L, Cheng H. FAEL: Fast autonomous exploration for large scale environments with a mobile robot. IEEE Robotics and Automation Letters. 2023;8(3):1667-74. doi:10.1109/LRA.2023.3236573.
24. Patel A, Saucedo MAV, Kanellakis C, Nikolakopoulos G. STAGE: Scalable and traversability-aware graph-based exploration planner for dynamically varying environments. In: 2024 IEEE Inter Conf Robot Autom (ICRA). 2024. p. 5949-55. doi:10.1109/ICRA57147.2024.10610939.
25. Cao C, Zhu H, Ren Z, Choset H, Zhang J. Representation granularity enables time-efficient autonomous exploration in large, complex worlds. Science Robotics. 2023 Jul;8(80):eadf0970. doi:10.1126/scirobotics.adf0970.
26. Wurm KM, Stachniss C, Burgard W. Coordinated multi-robot exploration using a segmentation of the environment. In: 2008 IEEE/RSJ Int Conf Intel Robot Syst (IROS). 2008. p. 1160-5. doi:10.1109/IROS.2008.4650734.
27. Thrun S. Learning metric-topological maps for indoor mobile robot navigation. Artificial Intelligence. 1998;99(1):21-71. doi:10.1016/S0004-3702(97)00078-7.
28. Luperto M, Kucner TP, Tassi A, Magnusson M, Amigoni F. Robust structure identification and room segmentation of cluttered indoor environments from occupancy grid maps. IEEE Robot Autom Lett. 2022;7(3):7974-81. doi:10.1109/LRA.2022.3186495.
29. Luperto M, Kucner TP, Tassi A, Magnusson M, Amigoni F. Mapping beyond what you can see: Predicting the layout of rooms behind closed doors. Robot Auton Syst. 2023;159:104282. doi:10.1016/j.robot.2022.104282.
30. Dharmadhikari M, Alexis K. Semantics-aware exploration and inspection path planning. In: 2023 IEEE Inter Conf Robot Autom (ICRA). 2023. p. 3360-7. doi:10.1109/ICRA48891.2023.10160469.
31. Long S, Li Y, Wu C, Xu B, Fan W. HPHS: Hierarchical planning based on hybrid frontier sampling for unknown environments exploration. In: 2024 IEEE/RSJ Int Conf Intel Robot Syst (IROS). 2024. p. 12056-63. doi:10.1109/IROS58592.2024.10802539.
32. Long S. HPHS planner [Internet]. GitHub Repository. 2024 [cited 2025 Jan 2]. Available from: https://github.com/bit-lsj/HPHS. Commit a96017c.
33. Galceran E, Carreras M. A survey on coverage path planning for robotics. Robot Auton Syst. 2013;61(12):1258-76. doi:10.1016/j.robot.2013.09.004.
34. Tan CS, Mohd-Mokhtar R, Arshad MR. A comprehensive review of coverage path planning in robotics using classical and heuristic algorithms. IEEE Access. 2021;9:119310-42. doi:10.1109/ACCESS.2021.3108177.
35. Choset H. Coverage of unknown spaces: the Boustrophedon cellular decomposition. Auton Robots. 2000;9(3):247-53. doi:10.1023/A:1008958800904.
36. Song J, Gupta S. ε*: An online coverage path planning algorithm. IEEE Trans Robot. 2018;34(2):526-33. doi:10.1109/TRO.2017.2780259.
37. Guruprasad KR, Ranjitha TD. CPC algorithm: Exact area coverage by a mobile robot using approximate cellular decomposition. Robotica. 2021;39(7):1141-62. doi:10.1017/S026357472000096X.
38. Song S, Kim D, Jo S. Online coverage and inspection planning for 3D modeling. Auton Robots. 2020;44:1431-50. doi:10.1007/s10514-020-09936-7.
39. Beeson P, Jong N, Kuipers B. Towards autonomous topological place detection using the extended Voronoi graph. In: 2005 IEEE Inter Conf Robot Autom (ICRA). 2005. p. 4373-9. doi:10.1109/ROBOT.2005.1570793.
40. Y Yang W, Wang X, Farhadi A, Gupta A, Mottaghi R. Visual semantic navigation using scene priors. In: 7th International Conference on Learning Representations (ICLR). 2019 May 6-9.
41. Roberts M, Shah S, Dey D, Truong A, Sinha S, Kapoor A, Hanrahan P, Joshi N. Submodular trajectory optimization for aerial 3D scanning. In: 2017 IEEE International Conference on Computer Vision (ICCV). 2017. p. 5334-43. doi:10.1109/ICCV.2017.569
42. Cao C. TARE planner [Internet]. GitHub Repository. 2024 [cited 2025 Jan 2]. Available from: https://github.com/caochao39/tare-planner. Commit 445007.

CONTRIBUTORS

Mr Nitin Kumar Dhiman obtained his BTech (CSE) from Kurukshetra University. He is currently working as a Scientist 'F' at the DRDO-CAIR, Bengaluru. His research interests include: Cognitive robotics.
In the current study he conceptualized the approach, software implementation, draft creation, result validation and simulation studies.

Mr Aniket Kalra is pursuing a BS in Data Science and Applications from IIT Madras, Chennai and currently working as Technician 'A' at DRDO-CAIR, Bengaluru. His areas of interest include: Robotics, deep learning and computer vision.
In the current study he contributed in software development, simulation setup and testing.

Mr K. Dali Naidu obtained his MTech (Control) from IIT Bombay and currently working as Scientist 'D' at the DRDO-CAIR, Bengaluru. His areas of interest include: Path planning and motion planning for autonomous robots.
In the current study he contributed in behaviour-tree-based control flow implementation of the algorithm.

Prof. (Dr) Deepak Khemani obtained his PhD from IIT Bombay

and currently working as a Professor at Plaksha University. He retired as a Professor from Dept. of CSE, IIT Madras. His research interests include: Search methods for problem-solving, knowledge representation, memory-based reasoning, constraint satisfaction, game-playing and epistemic logic.
His contributions towards this research include: Draft checking, supervision and overall guidance.