AIAA SciTech Forum
8-12 January 2024, Orlando, FL
AIAA SCITECH 2024 Forum

10.2514/6.2024-2017

Reachabilty Analysis of UAVs through Cluttered Environments

Kumar Ankit *, Shrivathsa M S †, Shishir Kolathaya ‡ and Debasish Ghose§
Indian Institute of Science, Bangalore

Aerial imaging and surveillance play a crucial role in forestry and agriculture processes, utilizing Unmanned Aerial Vehicles (UAVs) for high-resolution remote sensing at close range. However, navigating through dense canopies or underneath them without colliding remains a challenging problem in aerial robotics. Quantifying the reachability of aerial space is a prerequisite for deciding upon the technical feasibility of such operations. Existing literature often suggests using reactive range-based methods that rely on expensive onboard sensors like depth cameras or LiDARs. This work presents a cost-effective geometrical analysis using a monocular camera and inertial sensors to generate reachability maps. An essential contribution is proposing an efficient sector division method that significantly reduces the search space required for analysis. The generated reachability map can serve as a valuable resource for planning optimal paths and operational settings, facilitating the collision-free traversal of UAVs through complex aerial environments with various obstructions and clutter.

I. Introduction

Unmanned Aerial Vehicles (UAVs) offer a significant advantage over ground vehicles for rapid exploration and mapping of outdoor environments. Despite rapid advances in GPS-denied localization [1], reactive navigation, [2] and collision avoidance strategies [3], safety remains a significant concern for cluttered environments such as forests or plantations. To address this challenge, our work proposes a cost-effective and energy-efficient approach based on a monocular camera-based geometrical analysis of aerial space.

Several factors affect the easy traversal of aerial robots through complex environments. A thorough analysis of these factors, such as visibility [4], wind [5], reachability [3], path length [6], provide insights on using specific sensors and algorithms to ensure safe operation. A critical aspect, the aerial reachability analysis of environments, has yet to be studied in detail. This work proposes a cost-effective and energy-efficient monocular camera-based geometrical analysis of aerial space proximal to a forest's canopy and the factors affecting aerial reachability. Through this analysis, certain constraints can be imposed on the UAVs to optimize motion planning tasks further.

Traversability of outdoor environments has been studied for rovers and other ground vehicles, with an aim to navigate autonomously through complex unstructured environments [7]. These analyses focus on learning-based approaches that distinguish between navigable and non-navigable terrain using 3D Lidar data [8]. Traversibility is also an important consideration in indoor environments [9].

This work aims to aid UAVs to traverse safely under the canopy through cluttered environments by first modelling the reachable region. Trees generally display variable cross-sections, bringing in visibility [4] and reachability variation [3] at different heights. Path planning in such environments with sampling-based methods such as RRT* [6] or Dijkstra's algorithm [10] needs to consider the variable horizontal cross-sections of trees at different altitudes with varying canopy densities [11]. Furthermore, the factors that cause a hindrance to UAV traversal need studying to optimize the energy spent by UAVs in long-haul missions.

Our proposed approach can also address some critical challenges government agencies and farmers face while monitoring forests or developing farms over time. For example, our approach can enable the creation of high-resolution maps that can be used to monitor changes in forests or farmland over time. This information can help government agencies, and farmers make informed decisions about land management practices. Our approach can also enable the capture of close-range photogrammetry, which is essential for monitoring the health of vegetation and phenotyping.

*PhD student, Robert Bosch Centre for Cyber-Physical Systems (RBCCPS), Indian Institute of Science, Bangalore, India, kumarankit@iisc.ac.in
†Intern at Robert Bosch Center for Cyber-Physical Systems RBCCPS, Indian Institute of Science, Bangalore, India, msshrivathsa@gmail.com
‡Faculty of RBCCPS and Department of Computer Science & Automation, Indian Institute of Science, Bangalore, India, shishirk@iisc.ac.in
§Faculty of RBCCPS and Department of Aerospace Engineering, Indian Institute of Science, Bangalore, India, dghose@iisc.ac.in

II. Problem Statement

The reachability of unmanned aerial vehicles refers to their capability to access and navigate specific areas within their operational range. It involves a comprehensive analysis and determination of feasible paths that UAVs can follow to reach their intended destinations or carry out designated tasks.

The evaluation of UAV reachability revolves around two crucial aspects. Firstly, it examines whether the physical dimensions of the UAV allow it to fit into a particular region. Secondly, it investigates the availability of a collision-free path that enables the UAV to reach the desired location without encountering obstacles. Only when both conditions are met can the location be classified as reachable; otherwise, it is deemed unreachable. This assessment considers various potential obstacles, such as buildings, trees, power lines, and other obstructions. These obstructions can potentially impede the UAV's flight path or pose risks of collision. The concept of reachability maps generated from obstacle maps is depicted in Figure 1. In this figure, the green region represents the areas that can be reached by the drone, considering its depicted size. On the other hand, the red region indicates the inaccessible areas that cannot be reached by the drone within the given size constraints.

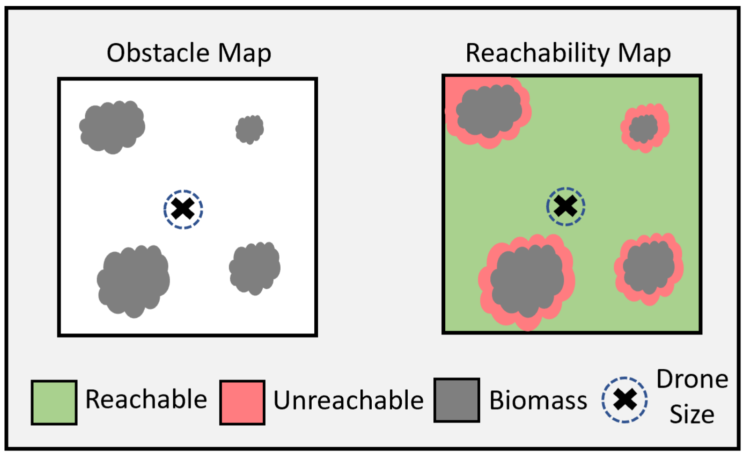


Fig. 1 Reachability of UAVs at an Operating Height

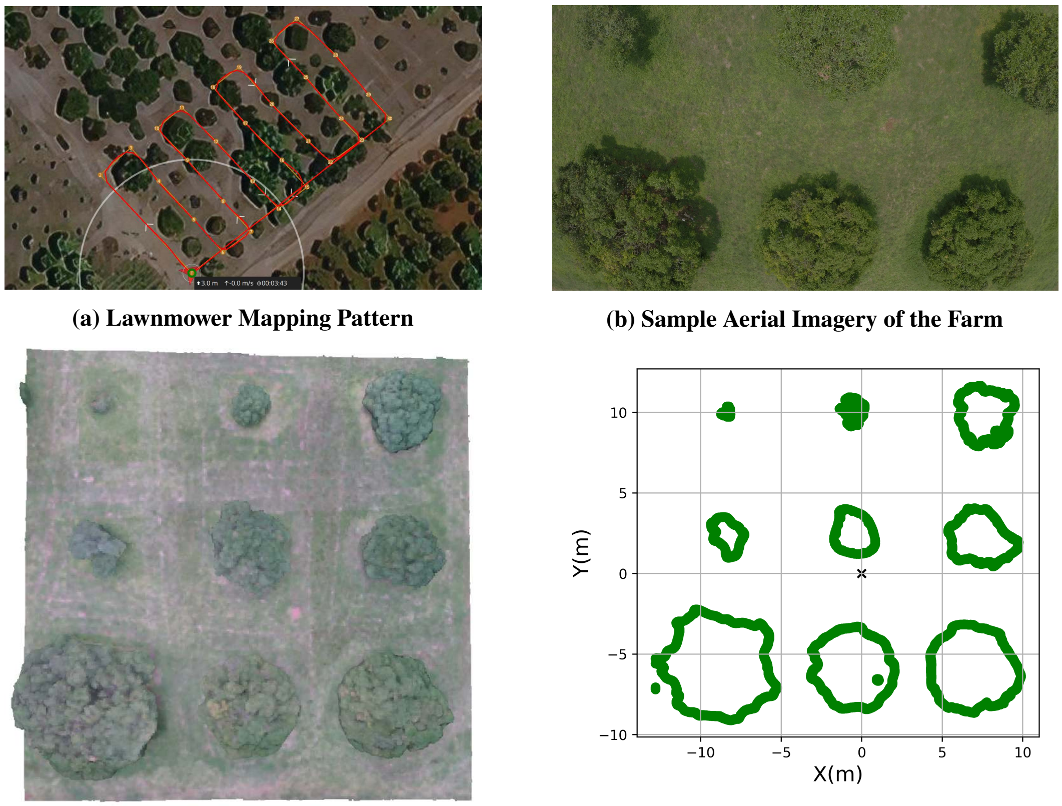
By understanding UAV reachability comprehensively, operators and mission planners are empowered to make informed decisions regarding flight routes, mission planning, and operational parameters. This knowledge is crucial in ensuring the successful execution of UAV missions while simultaneously prioritising safety, efficiency, and compliance with regulatory requirements. By accurately assessing the reachability of UAVs, operators can mitigate risks, optimise operational effectiveness, and ensure the overall success of UAV missions across diverse applications and industries. In this study, we conduct preliminary aerial mapping to create a model of the environment. Subsequently, we employ an iterative sector division technique to assess the reachability of different areas. The following sections comprehensively describe and analyse the methodology employed to generate these reachability maps.

III. Preliminary Aerial Mapping

In our assessment process, the first crucial step is to create a comprehensive 3D model of the operating environment. With limited prior knowledge about the environment, except for its boundaries, we begin with aerial mapping, which generates an initial representation of the environment as a sparse 3D point cloud.

To ensure precise mapping, we capture high-resolution RGB images of the canopy, considering the camera's footprint and capturing them in a lawnmower pattern with desired overlaps (Fig. 2a). These captured images are then processed using a popular Structure-from-Motion (SfM) software library named COLMAP[12], used for computer vision and photogrammetry tasks. This processing technique solves a large-displacement optical-flow problem and generates a sparse 3D point cloud from a set of 2D images. It involves analyzing the visual correspondences between images and leveraging the geometric constraints to estimate the camera poses and the 3D positions of scene points.

To refine the accuracy and reliability of the output point cloud, we further apply a radius-based outlier removal method. The radius-based outlier removal method examines each point in the point cloud and analyses its local neighbourhood density. A search radius is defined for each point, which determines the range within which neighbouring



(a) Lawnmower Mapping Pattern

(b) Sample Aerial Imagery of the Farm

(c) Generated Pointcloud

(d) Obstacle Map (at a height of 3m)

Fig. 2 Preliminary Aerial Mapping

points will be considered. Points with significantly lower neighbourhood densities than the surrounding points are considered potential outliers. Removal of these outliers results in a more precise and detailed point cloud that accurately depicts the environment. The resulting point cloud (Fig. 2c) is based on aerial images captured at the height of 10 meters, as depicted in Figure 2b. Subsequently, the point cloud is then sliced horizontally at a preferred operating height of the UAV, creating an obstacle map (see Fig. 2d) that accurately depicts the region at that specific elevation. A demonstration video of preliminary mapping can be found here.

IV. Sector Division Method

A straightforward approach to determine reachability would involve discretising the region into a uniform grid and exhaustively checking the reachability of each cell using a breadth-first search technique. However, we propose an iterative algorithm that reduces the number of path queries involved by leveraging QuadTrees. This is achieved by decreasing the amount of space that needs to be assessed for reachability.

The sector division method refers to the division of a given area or space into sectors or sub-regions. Our algorithm 1 divides the entire region into equal sub-regions to rapidly evaluate the reachability of empty or nearly empty sectors. In this study, we have introduced and examined the concepts of binary division and quadrant division. Furthermore, we have extended these principles to encompass divisions into parts that are positive integer exponents of 2, allowing for a generalised approach. In general, If a sub-sector is empty and a path exists to the centre of that sector, then the entire sub-sector can be deemed reachable. On the other hand, if the sub-sector contains obstacles, it is subdivided into further smaller sub-sectors. Then, a path is queried between the parent sector centre and centres of daughter sectors, and the algorithm repeats until the sector's size becomes smaller than the robot's dimensions.

Path-finding algorithms like Breadth-for-search or Dijkstra's algorithm can be employed to determine the paths between successive sector centres. By disregarding empty regions, the search space is significantly reduced, enabling a swift analysis of the reachability of the entire region. This approach is more efficient than a brute-force method, as it exploits QuadTrees and the concept of subdividing sectors to expedite reachability assessment.

In the upcoming sub-sections, we will present our methodology for two scenarios. The first scenario is when the region can be perfectly divided into the smallest possible grids, and the number of these grids is an integer exponent of 2 (such as 4, 8, 16, etc.). The second scenario pertains to regions that have a rectangular shape without the requirement of being perfectly divisible into the smallest grids. By covering both scenarios, we aim to provide a comprehensive understanding of our methodology for reachability analysis in different types of regions.

Algorithm 1 Reachability Analysis

```
Input: Obstacle List, Start Point
Store the points in a QuadTree data structure
 Traverse the tree breadth-first using a queue
 Check if the current node is occupied or free
 if a sector is unoccupied then
   if a path exists to the center of the parent of the node
    then
      Mark sector as Reachable
   else if a path can be found to the node from any of its ancestor nodes, or from the start point
    then
      Mark sector as Reachable
   else
      Mark sector as unreachable and add the node's children to the queue
else
   Add the node's children to the queue
end when
1) All the sectors are reachable OR
2) Quadtree node size is less than or equal to the size of the drone
```

A. For Perfectly Divisible Regions

A grid of size m × m and drone size of radius(r) is considered. The minimum grid size required for the analysis is determined by the drone's radius, which means that each grid should have dimensions of at least r × r to accommodate the drone. In cases where the region can be perfectly divided into these minimum grids, it implies that the ratio between the overall size of the region (m) and the drone's radius (r) should be a power of 2. This condition ensures that the region can be evenly divided into the smallest grids possible without any leftovers.

1) **Binary Division**: In this case, we divide the region into two halves by splitting along the shortest edge. If both edges have the same length, we choose any edge for the splitting process. Our initial step involves checking if either of these halves is occupied by querying the quadtree generated from the obstacle map. If we encounter an occupied region, we further divide it into two halves. We then search for a path that connects the centre of the parent sector to the centres of the newly divided halves. If a region is found to be unoccupied, it is considered reachable under the assumption that our assessment begins from the centre of the region. A sector is considered completely reachable if it is free from obstacles and the agent can reach the centre of the parent sector. This process continues until we have traversed the entire region or the daughter sectors become smaller than the drone's size. It is important to note that, during each division, two path queries are required, except for the final iteration, where only an occupancy check is needed.
   To evaluate the complexity of this division method, we have analyzed it in the worst-case scenario where the entire region is occupied. In such cases, the algorithm terminates once it has reached the minimum grid size and covered the entire area. The progression of the algorithm for binary divisions can be observed in Figure 3. The number of divisions (n_{div}) required before reaching the minimum grid size can be calculated as,

$$r = \frac{m}{2^{n_{div}/2}} \implies 2^{n_{div}/2} = \frac{m}{r} \implies n_{div} = 2\log_2\left(\frac{m}{r}\right) \tag{1}$$

   As mentioned above, in the above equations, n_{div} is an even number. Total number of path queries (n_{path}) required until minumum grid size is reached can be summed up to,

$$n_{path} = 2 + 4 + 8 + 16 + \ldots \tag{2}$$
$$= 2^1 + 2^2 + 2^3 \cdots + 2^i \tag{3}$$
$$= \sum_{i=0}^{n_{div}-1} 2^i = \left(2^{n_{div}} - 2\right) \qquad \text{where, } n_{div} = 2\log_2(m/r) \tag{4}$$

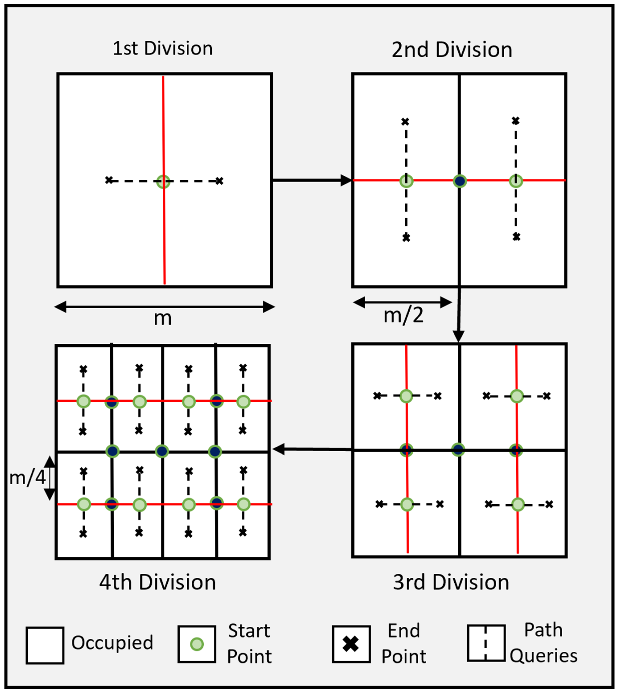


Fig. 3 Sector Division into Halves and Path Queries

2) **Quaternary Division**: Similarly, in this division, the considered region is split into four equal parts by drawing two perpendicular lines that intersect at the centre of the square. This results in four quadrants, each occupying one-fourth of the total area. For each quadrant, we initially determine if it is occupied or unoccupied. If it is unoccupied, we consider the sector reachable, assuming we start our assessment from the region's centre. However, if the quadrant is occupied, we divide it into smaller daughter quadrants. We then check the feasibility of a path to the centre of these daughter quadrants starting from the previous centre point. In this case, we need to search for four paths for each quadrant division.
We have analysed the algorithm's computations, considering a worst-case scenario where the region is fully occupied with obstacles. In such a case, the algorithm will terminate when it has covered the entire area and reached the minimum grid size. The algorithm's progression can be visualised using a graphical representation, as shown in Fig. 4. The number of divisions (n_{div}) required before reaching the minimum grid size, in this case, will be,

$$r = \frac{m}{2^{n_{div}}} \implies 2^{n_{div}} = \frac{m}{r} \implies n_{div} = \log_2\left(\frac{m}{r}\right) \tag{5}$$

The total number of path queries (n_{path}) required until minimum grid size is reached is,

$$n_{path} = 4 + 16 + 64 + \ldots \tag{6}$$
$$= 4^1 + 4^2 + \cdots + 4^i \tag{7}$$
$$= \sum_{i=0}^{n_{div}-1} 4^i = \left(\frac{4^{n_{div}} - 4}{3}\right) \qquad \text{where, } n_{div} = \log_2(m/r) \tag{8}$$

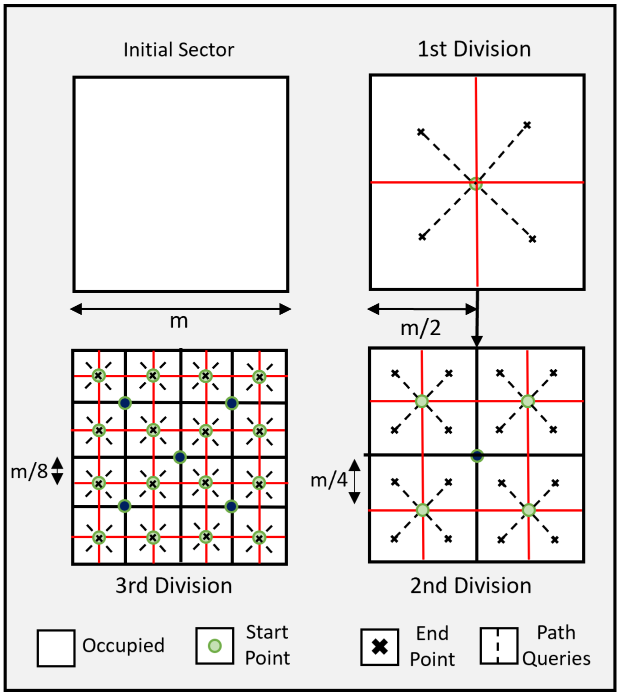


Fig. 4 Sector Division into Quarters and Path Queries

3) **General Division into Exponents of 2**:Using the principle of mathematical induction for several divisions pertaining to integer exponents of 2, a generalised expression for any such divisions can be calculated. Therefore, the total number of divisions required before reaching the minimum grid size is given as,

$$n_{div} = 2\log_{base}\left(\frac{m}{r}\right) \tag{9}$$

Number of path queries required to completely cover the entire region is,

$$n_{path} = \left(\frac{(base+1)^{n_{div}} - 1}{base}\right) \tag{10}$$

Since computation depends upon the base chosen, the highest base (a power of 2) is chosen such that $base \leq m$.

B. For Rectangular Regions

For a rectangular case, we consider a grid with dimensions m × n, where m represents the number of rows and n represents the number of columns. Similar to the previous explanation, we also have a drone with a specified radius (r) that determines the minimum grid size required for the analysis, which is r × r. To ensure proper division of the rectangular grid into minimum grids, the ratios m/r and n/r should be integers. However, in cases where these ratios are not integers, we consider an adjusted grid size with dimensions m' × n', where m'/r and n'/r are both integers. This adjustment allows us to maintain a consistent and accurate analysis by ensuring the grid can be divided without any remaining areas while still accommodating the specific dimensions and drone radius involved. Further, the region is partitioned into two sections: a square region that can be perfectly divided into smaller grids, and another region that encompasses marginal cells of the remaining area as depicted in figure below. Each of these parts is analyzed separately, as described in the following sections.

(i) **Square Region:** This part specifically refers to the largest square sector that can be formed within the region, with its size being a power of 2, which is depicted in yellow colour in Fig. 5. This is being solved using the sector division method stated in Algorithm 1. The number of divisions required to reach the minimum grid size is determined by Equation 9, which takes into account the base chosen for the division process. Additionally, during each division, path queries are performed to assess the reachability within the sectors.

The number of path queries required is determined by Equation 10, again based on the base chosen. These path queries involve evaluating the connectivity and accessibility of the central points of the parent sector to the central points of the newly divided sub-sectors.

(ii) **Marginal Cells:** This part includes the remaining portion of the grid, depicted in blue in Figure 5. We solve this part by discretizing it into the smallest possible grid size, which is equal to the size of the UAV. Each of these grids is then individually checked using the Breadth-first Search algorithm to determine occupancy and to assess the feasibility of a path, thus determining the reachability. By analyzing the connectivity of the neighbouring cells and assessing the absence of obstacles or occupied areas, we can determine if a path can be established from one cell to another within the grid. For the second case, the number of path (n_{path}) queries will be,

$$n_{path} = m^2 - base^2 \tag{11}$$

Therefore, the total number of path queries (n_{path}), in this case becomes,

$$n_{path} = \left(\frac{(base+1)^{n_{div}} - 1}{base} \right) + (m^2 - base^2) \tag{12}$$

where the number of divisions n_{div} required to assess the first part depends upon the region's dimensions and the base chosen (refer to Eq.9).

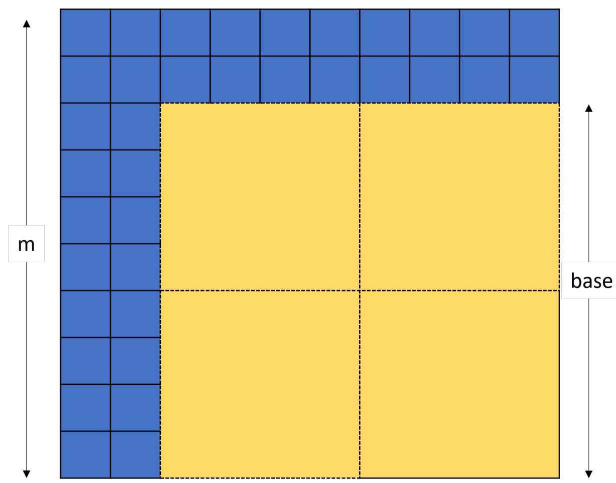


Fig. 5 General Square Sector Partition

V. Reachability Analysis Results

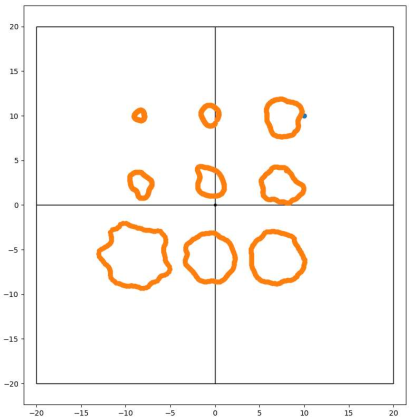
We performed the reachabilitity analysis on the selected region, a farm which consists of trees planted in a grid. Aerial Images of the farm were captured and by the process of 3D reconstruction, aided by the COLMAP program, the pointcloud represented in Figure 2c was obtained. An operating height for the UAV was decided, and using the slicing operations described earlier, a 2D obstacle map was obtained.

The reachability analysis is performed on the generated obstacle map. The region of operation is taken to be a square of dimensions 40m x 40m. The algorithm used for the analysis is based on the approach explained in the previous section. The algorith was implemented in Python. First, the obstacle information extracted from the map is stored in a QuadTree data structure. Additionally, we assume that the UAV has dimensions of 1m x 1m.

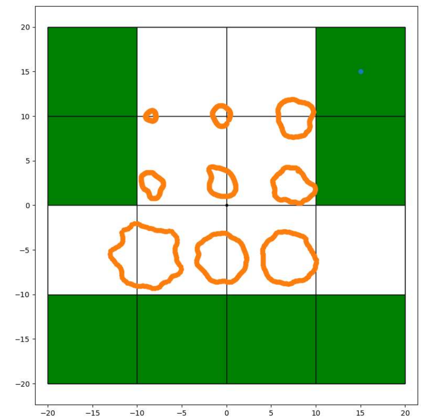
In each iteration of the algorithm, the occupied regions in the obstacle map are recursively divided into quarters. Starting from the parent sector which consists of the entire region of operation, every sector is checked for presence of obstacles using the QuadTree. This division process is demonstrated in Fig.6, illustrating how the algorithm segments the regions occupied by obstacles. We utilize Dijkstra's algorithm to find the path from centre of parent sector to the centres of daughter quadrants. The map visually represents the progression of the algorithm over five iterations.

Once the analysis is completed in the fifth iteration, the reachability map is generated, as shown in Figure 6d. The reachability map indicates which areas of the region are reachable for the UAV. In this case, most of the map is determined to be reachable due to the relatively sparse distribution of obstacles. The areas enclosed by clusters of obstacles are identified as unreachable regions. An animation of the analysis can be found here.

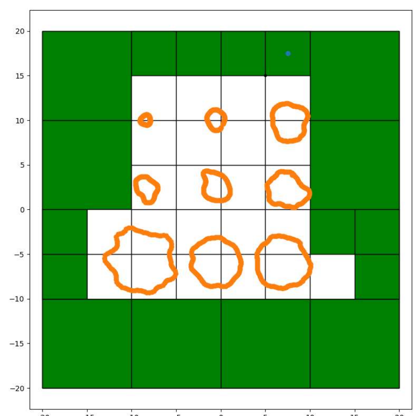
To quantify the reachability, we assess the fraction of inaccessible area by comparing the size of these sectors to the total region. This allows us to estimate the obstacle density. In the case of the plantation environment, the obstacle density was determined to be **14%**. It is important to note that this value is specific to the height at which the obstacle map was generated and may vary accordingly. In other words, the available space for the UAV to navigate differs at different heights. By utilizing the concepts presented in this paper, users can make informed decisions regarding the operating altitude based on the specifications of the agent and the desired task.



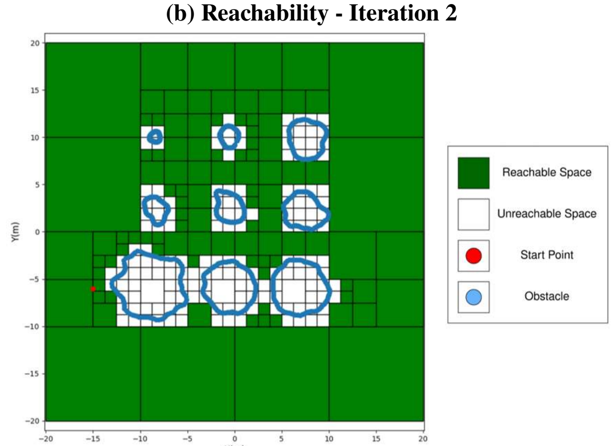
(a) Reachability - Iteration 1



(b) Reachability - Iteration 2



(c) Reachability - Iteration 3



(d) Final Iteration

Fig. 6 Reachability Results

VI. Conclusion & Future Works

The key methodology employed is the sector division method, which allows for efficient analysis by disregarding empty regions and significantly reducing the search space. The algorithm minimises the number of path queries required by partitioning the environment into sectors and selectively considering occupied areas.

The work delves into two specific types of sector division: binary and quadrant division. These division methods are thoroughly explained, providing insights into how they contribute to the overall reachability analysis process. Furthermore, the research extends the division techniques to a more generalised method, allowing for flexibility in handling rectangular regions. In upcoming publications, it will also be extended to areas with arbitrary curvilinear boundaries and 3D cases.

Acknowledgement

This work is done under the Prime Minister's Research Fellowship (PMRF) scheme, Ministry of Education, Government of India.

References

[1] Chisholm, R. A., Cui, J., Lum, S. K., and Chen, B. M., "UAV LiDAR for below-canopy forest surveys," *Journal of Unmanned Vehicle Systems*, Vol. 1, No. 01, 2013, pp. 61–68.

[2] Martínez, J. L., Morales, J., Sánchez, M., Morán, M., Reina, A. J., and Fernández-Lozano, J. J., "Reactive Navigation on Natural Environments by Continuous Classification of Ground Traversability," *Sensors*, Vol. 20, No. 22, 2020, p. 6423.

[3] Zhou, Y., and Baras, J. S., "Reachable set approach to collision avoidance for uavs," *2015 54th IEEE Conference on Decision and Control (CDC)*, IEEE, 2015, pp. 5947–5952.

[4] Klodt, L., Khodaverdian, S., and Willert, V., "Motion control for UAV-UGV cooperation with visibility constraint," *2015 IEEE Conference on Control Applications (CCA)*, IEEE, 2015, pp. 1379–1385.

[5] Patrikar, J., Moon, B. G., and Scherer, S., "Wind and the city: Utilizing uav-based in-situ measurements for estimating urban wind fields," *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, IEEE, 2020, pp. 1254–1260.

[6] Karaman, S., and Frazzoli, E., "Sampling-based algorithms for optimal motion planning," *The international journal of robotics research*, Vol. 30, No. 7, 2011, pp. 846–894.

[7] Kim, D., Sun, J., Oh, S. M., Rehg, J., and Bobick, A., "Traversability classification using unsupervised on-line visual learning for outdoor robot navigation," *Proceedings 2006 IEEE International Conference on Robotics and Automation, 2006. ICRA 2006.*, 2006, pp. 518–525. https://doi.org/10.1109/ROBOT.2006.1641763.

[8] Suger, B., Steder, B., and Burgard, W., "Traversability analysis for mobile robots in outdoor environments: A semi-supervised learning approach based on 3D-lidar data," *2015 IEEE International Conference on Robotics and Automation (ICRA)*, 2015, pp. 3941–3946. https://doi.org/10.1109/ICRA.2015.7139749.

[9] De Croon, G., and De Wagter, C., "Challenges of Autonomous Flight in Indoor Environments," *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2018, pp. 1003–1009. https://doi.org/10.1109/IROS.2018.8593704.

[10] Dijkstra, E. W., "A note on two problems in connexion with graphs," Springer, 1959, pp. 269–271.

[11] "Forest Cover mapping - Scheme of Classification," , 2020. URL https://fsi.nic.in/introduction.

[12] Schonberger, J. L., and Frahm, J.-M., "Structure-from-Motion Revisited," *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.