# Baseline Solution for Cooperative Aerial Inspection Challenge - Technical Report

Xinhang Xu, Muqing Cao, Thien-Minh Nguyen*, *Member, IEEE*

***Abstract*— In this paper we propose a neat solution to the cooperative aerial inspection. The solution centers on three key elements: mapping, task assignment, and cooperative policy. The key to this challenge lies in effectively coordinating a group of aerial robots to collaboratively explore and inspect intricate environments, utilizing prior information such as the bounding boxes of the interest area. In this approach, the explorers and photographers are allocated into teams, and the allocation of tasks is realized by assigning an appropriate target inspection volume to each team using best-first search. A voxel map-based representation of the environment is used for pathfinding. Using the proposed approach, a noteworthy achievement of 812.354 points is attained in the MBS simulation scene within a timeframe of 852.585 seconds.**

## I. Links

The baseline solution can be found at `https://github.com/ntu-aris/caric_baseline` for registerred participants.

The CARIC challenge website is announced at `https://ntu-aris.github.io/caric/`

## II. System Overview

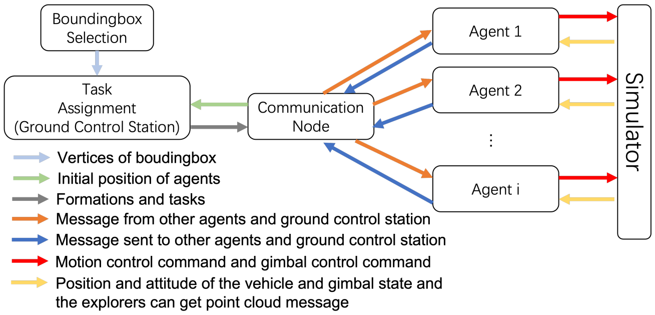

Boundingbox Selection
Task Assignment (Ground Control Station)
Communication Node
Agent 1
Agent 2
⋮
Agent i
Simulator
Vertices of boudingbox
Initial position of agents
Formations and tasks
Message from other agents and ground control station
Message sent to other agents and ground control station
Motion control command and gimbal control command
Position and attitude of the vehicle and gimbal state and the explorers can get point cloud message

Fig. 1: The system overview

The overall system architecture is shown in Figure 1. Initially, given the initial positions of all in-task agents and the coordinates of the bounding boxes containing all structures of interest, the task assignment module at the ground station assigns tasks and communicates them to all agents.

Following that, the agents embark on exploration and inspection activities in accordance with the pre-assigned tasks. Over time, they adapt and adjust their plans based on the additional environmental information they perceive, progressively working towards the fulfillment of the task.

## III. Task Assignment

### A. Team Assignments

The inspection task allows using two types of aerial robots, namely the explorer and the photographer. In the challenge guideline, only the explorer is equipped with a rotating 3D Lidar to fully perceive the external environments, while both the explorer and photographer carry gimbal-stabilized cameras for inspection. A group-based approach is designed where explorers and photographers are allocated into teams to expedite the process of exploration and inspection of multiple regions.

In this context, the ground station initially identifies the quantity of explorers present within the environment. Teams are formed where each team consists of a single explorer as the leader. Then explorers are assigned to teams based on the Euclidean distance to the respective explorers.

### B. Task Assignment

Following the team assignments, the ground control station proceeds to reconstruct the detailed bounding box information, including the center point, dimensions, and orientation with respect to the world frame. This reconstruction is derived from the vertices of the bounding boxes provided by the challenge organizer.

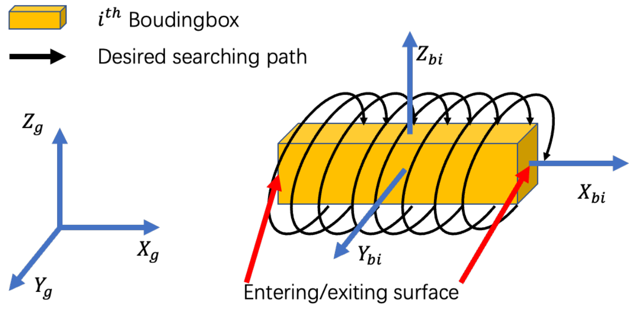


Fig. 2: Example of how to describe the $i^{th}$ boundingbox

We adopt an intuitive and efficient exploration strategy where an explorer follows a spiral trajectory around the axis parallel to the longest side of a bounding box, as illustrated in Figure 2. An exploring robot starts and ends the trajectory from the centers of the surfaces parallel to the Y-Z plane, which are called the entering/exiting points of a box. The direction of an exploration trajectory is therefore uniquely defined by the entering and exiting points.

The authors are with School of Electrical and Electronic Engineering, Nanyang Technological University, Singapore 639798, 50 Nanyang Avenue. Email: `xu0021ng@e.ntu.edu.sg`, `mqcao@ntu.edu.sg`, `thienminh.nguyen@ntu.edu.sg`.

Author to which correspondence should be made.

The allocation of bounding boxes among teams of robots is carried out using a simple best-first algorithm. Starting from the team with the highest member count, a bounding box with the nearest entering/exitting point to the explorer is assigned to this team, and the spiral trajectory is defined. An iterative process is conducted where the next closest bounding box to the end of the current trajectory will be assigned. To ensure balanced allocation, the total volume of the bounding boxes assigned to a team should be proportional to the number of team members. A bounding box may be divided into smaller ones if the sharing inspection volume among teams improves fairness. In this way, each team is tasked to explore a set of non-overlapping bounding boxes with distinct exploration paths.

## IV. Mapping

To ensure a balance between path planning safety and speed, we employ a low-resolution voxel map. The map's coarse granularity expedites planning while ensuring a large safety distance.

To reduce memory usage and improve search efficiency, the solution adopts a dual-map structure. This encompasses a shared global map alongside individual local maps of specific bounding boxes. All maps are built on voxel grids with a grid size of 2.5 meters—equivalent to 2.5 times the explorer's collision radius.

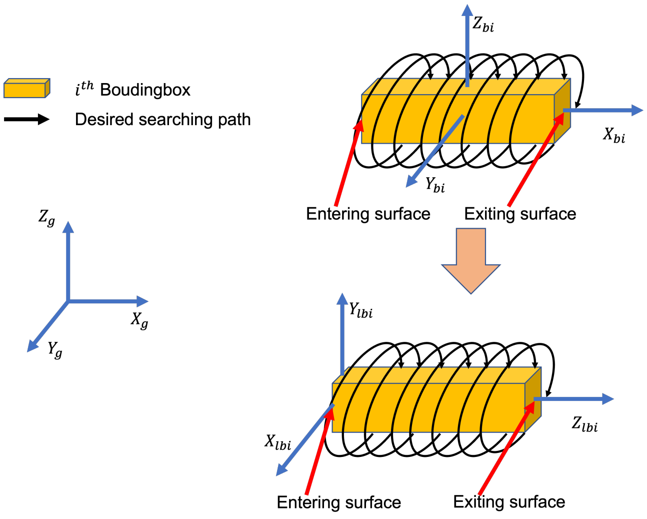


Fig. 3: Example of the $i^{th}$ boundingbox local frame

An intuitive way to cover the surface of a rectangular block is to use a spiral trajectory around the block. In an effort to achieve a smooth and intuitive path planning, a coordinate transformation is introduced in Figure 3, where the origin of the transformed coordinates is shifted to the center of the entering surface, with the Z axis pointing towards the center of the exiting surface. The axes with subscripts $lbi$ indicate the transformed axes of the bounding box coordinate system. This representation facilitates the subsequent comprehensive inspection process.

In occupancy map, each voxel is labeled with one of the three possible distinct states during the map construction process: occupied, interesting, and visited. A voxel containing a point from the scanned point cloud will be marked as occupied, and its adjacent six voxels will be labeled as interesting (to be visited). When an agent enters an interesting voxel and utilizes its gimbal to scan all the neighboring occupied voxels, the interesting voxel transitions into a visited state. The process is elaborated in detail in Figure 4.

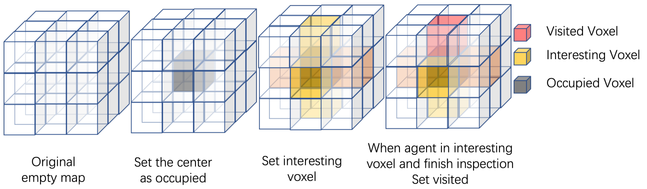

Visited Voxel
Interesting Voxel
Occupied Voxel
Original empty map
Set the center as occupied
Set interesting voxel
When agent in interesting voxel and finish inspection Set visited

Fig. 4: The voxel edition workflow

## V. Exploration and Inspection

There are two modes for each team in the task: the Transfer Mode and the Exploration and Inspection Mode. This section outlines the details of these two modes.

### A. Transfer Mode of explorer

The Transfer Mode is the phase in which the team navigates to the target bounding box area before the exploration and inspection, and subsequently returns to their initial positions upon task completion. During this mode, the explorer functions as the leader while the photographers follow the leader's path. The explorer initially uses A* algorithm to search for a feasible path to an unoccupied voxel on the entering surface of the target bounding box. The point identified during this search becomes the designated 'entering voxel'. The flowchart depicting the update process for the 'entering voxel' is presented in Figure 5.

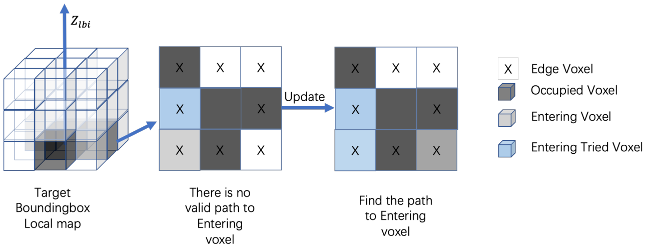

$Z_{lbi}$
X
Update
Edge Voxel
Occupied Voxel
Entering Voxel
Entering Tried Voxel
Target Boundingbox Local map
There is no valid path to Entering voxel
Find the path to Entering voxel

Fig. 5: Example of updating entering voxel

Subsequently, the explorer follows a predetermined path to the entering voxel. In the event that the path is found to be infeasible due to dynamic environments or map updates, the explorer will try to rediscover a suitable path. If this proves unfeasible, the entering voxel is updated accordingly. This adjustment is communicated to the following photographers through the dedicated communication node, ensuring the entire team is well-informed and adaptable to map update.

### B. *Transfer Mode of photographer*

During the Transfer Mode, the photographer continuously tracks the explorer's movements. Once the explorer exits the Transfer Mode, the photographer navigates to the designated fly-in voxel and awaits the explorer's subsequent commands. This coordinated interaction ensures that the photographer aligns seamlessly with the explorer's actions, maintaining efficient and secure navigation throughout the transition from Transfer Mode to subsequent operations.

### C. *Exploration and Inspection Mode*

Given the strong compatibility between the explorer and photographer during the Exploration and Inspection Mode, this subsection provides an integrated overview of their collaborative functions.

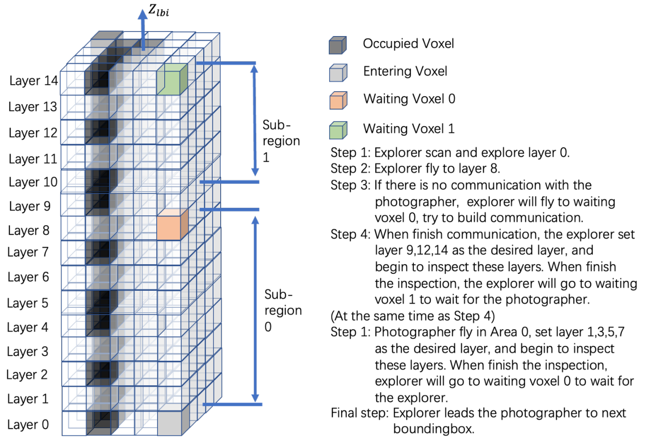
In each bounding box assigned to a team, the volume is further divided into multiple sub-regions, one for each agent. The explorer will first conduct a quick mapping of the other agents' subregions, before going to its assigned sub-region to do a complete scan. Given that the Explorer has a slower speed and needs to build the map prior to the inspection, the Explorer will be assigned a smaller task area and the area it needs to perform a complete scan will always be at the farthest end of the bounding box's search direction. The task assignment is shown in Figure 6.

Once the explorer is in the entering voxel, a transition from Transfer Mode to Exploration and Inspection Mode occurs. Within each segmented task interval, the explorer undertakes a sequence of actions:

- Quick mapping at the bottom layer of a subarea: To obtain a point cloud map of a subregion, the strategy used is to fly the explorer around the bottom layer of each subregion. To do so, the explorer initiates a Dijkstra search on the bottom layer of the task area. This search aims to identify the nearest interesting but unvisited voxels within the bottom layer. If the bottom layer lacks occupied voxels, the search returns the voxel at the boundary that remains unvisited in this layer.
- Communicating the map to the photographer: Upon successful completion of the mapping, the explorer proceeds to the bottom layer of the next task layer. Simultaneously, the explorer actively engages in communication with the assigned photographer for this task subregion. The photographer is informed of the readiness to commence area inspection.
- Communication handling: In instances where communication with the photographer cannot be established, the explorer embarks on an adaptive strategy. It first navigates to the same x and y coordinates as the photographer, which waits at the entering voxel . If this approach is unsuccessful, the explorer continues by moving to the adjacent free voxels.

This series of actions enables the explorer to synchronize with the photographer and accommodate communication challenges that may arise during the process.

Upon finishing mapping of the other agents' subregion, the explorer proceeds to conduct a detailed scanning of the final subregion. Based on the provided camera FOV and score metrics, to conduct a full scanning, the explorer needs to inspect one in every three layers. To scan a target layer, the explorer initiates a Dijkstra search to an unvisited but interesting voxel situated at that specific layer. If the search fails to find a viable path, the explorer progresses to the next target layer. Upon completing all assigned tasks, the explorer navigates to the nearest voxel aligned with the entering point's x-index and y-index for that task layer. It then assumes a waiting state.



Step 1: Explorer scan and explore layer 0.
Step 2: Explorer fly to layer 8.
Step 3: If there is no communication with the photographer, explorer will fly to waiting voxel 0, try to build communication.
Step 4: When finish communication, the explorer set layer 9,12,14 as the desired layer, and begin to inspect these layers. When finish the inspection, the explorer will go to waiting voxel 1 to wait for the photographer.
(At the same time as Step 4)
Step 1: Photographer fly in Area 0, set layer 1,3,5,7 as the desired layer, and begin to inspect these layers. When finish the inspection, explorer will go to waiting voxel 0 to wait for the explorer.
Final step: Explorer leads the photographer to next boundingbox.

Fig. 6: The example of a team with one explorer and one photographer finish one assigned boundingbox

For the photographers, upon receiving the command indicating that the explorer has mapped the task subregion, the photographer undertakes a series of actions as following steps:

- Entry through the entering Voxel: The photographer, prompted by the command, enters the designated task area via the entering voxel. As it enters this region, the photographer's operational mode shifts from Transfer Mode to Exploration and Inspection Mode.
- Layered scanning: Similar to the explorer, within the task area, the photographer organizes the scanning process in layers, where scanning layers are established at intervals of three layers. The photographer employs the Dijkstra algorithm to scan each layer thoroughly.
- Transition to waiting State: Upon completing the scan of all designated layers, the photographer proceeds to navigate to the nearest voxel that shares the same x-index and y-index as the entering voxel. In this location, the photographer assumes a waiting state, anticipating further instructions from the explorer.

Upon completion of tasks by all photographers within the team's designated task interval, the explorer commands all team members to transition into Transfer Mode. The team collectively takes flight towards the next task bounding box area. The example of a team with one explorer and one photographer finishing one assigned bounding box is shown in Figure 6.

### D. Collision Avoidance Rules

To ensure effective collision avoidance, all agents within the system publish their current global positions and future positions according to the planned paths. This information is shared among the agents and serves as a reference during their path-planning processes. Different obstacle avoidance priorities are assigned to agents based on their roles and the alphabetical order of their names to prevent deadlock issues arising from mutual obstruction between agents. This collaborative approach enhances the overall safety and efficiency of the system's operations.

## VI. Example of Workflow

To facilitate understanding, a complete workflow is introduced here through an example task including two explorers and three photographers. In the beginning, the ground station assigns the bounding boxes among two teams, as shown in Figure 7.

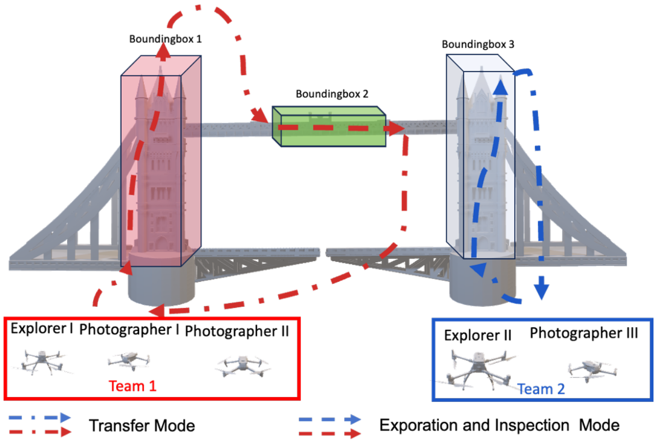


Fig. 7: The example of the ground station assign the task

We focus on the workflow of Team 2 while Team 1's behaviour is the similar. First, the bounding box is divided into different task areas for the explorer and the photographer. Then explorer II searches for the entering voxel and a path to reach the voxel. At the same time, photographer III follows explorer II until it is in the entering voxel, as illustrated in Figure 8.

In the bounding box, the explorer first explores the photographer's task area and then shares the map with the photographer, who starts the inspection thereafter. At the same time, the explorer conducts the inspection in its task area. When completing the inspection of the current bounding box, the team flies to the next assigned bounding box or, in the case that all boxes are inspected, goes back to the starting positions, as illustrated in Figure 9.

## VII. Result

A noteworthy achievement of 812.354 points is attained within the context of the mbs subtask, all realized within a timeframe of 852.585 seconds.

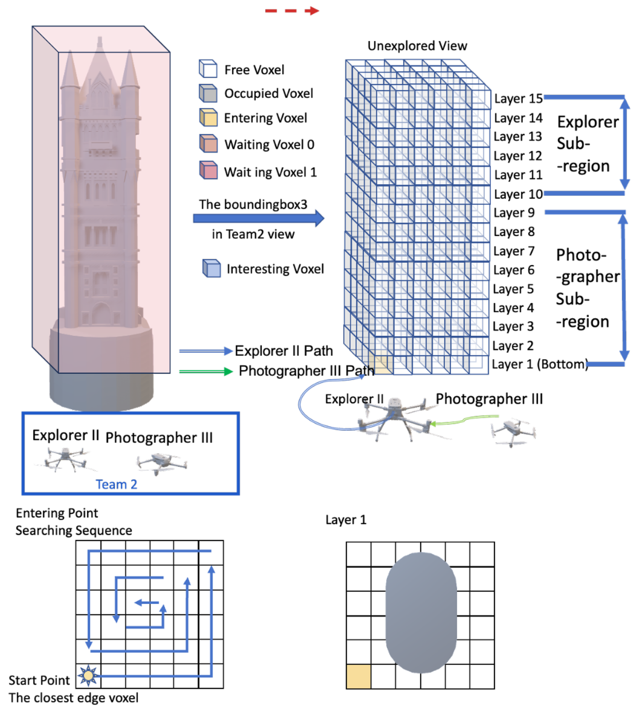


Fig. 8: The example of how a team enters the boundingbox

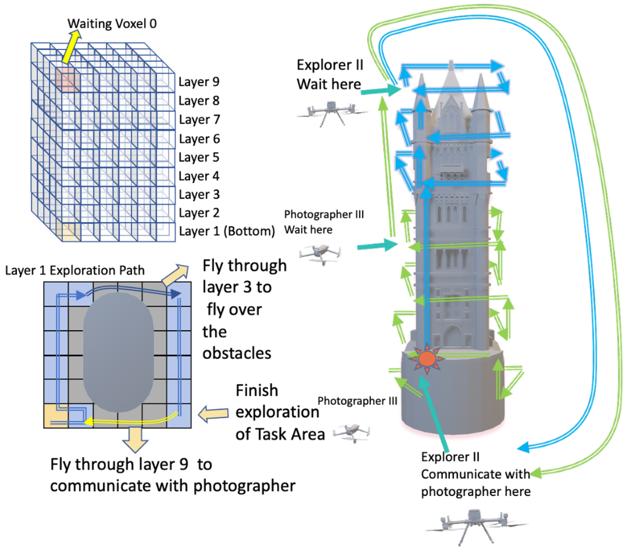
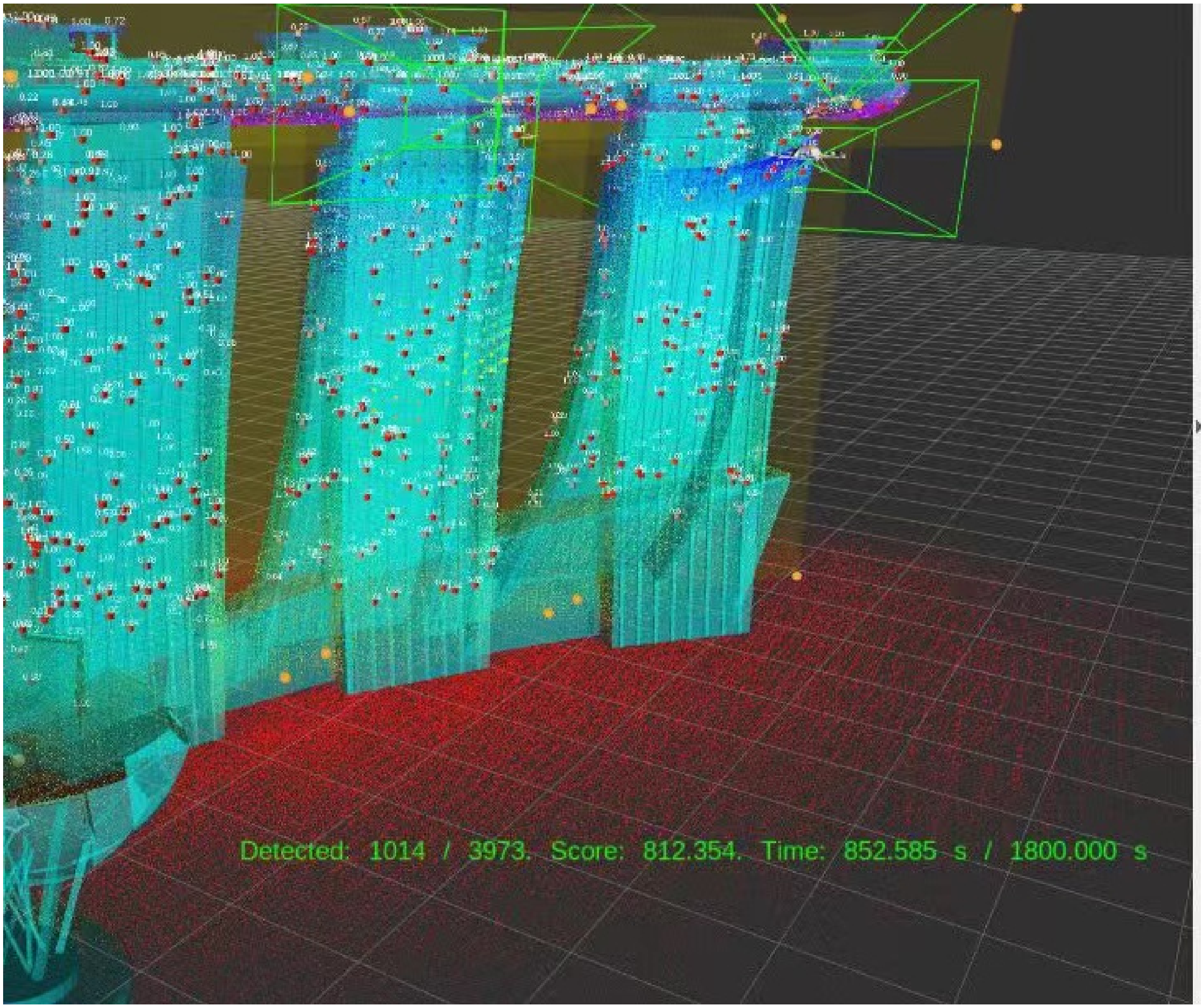


Fig. 9: The example of exploration and inspection



```
Detected:  1014  /  3973.  Score:  812.354.  Time:  852.585  s  /  1800.000  s
```

Fig. 10: The result of mbs subtask