Heterogeneous Map Fusion Method of Generalized Voronoi Diagram (GVD) and OctoMap Based on Point Cloud

Yinuo Kang[1[0009-0004-1428-6182]], Lan Cheng[1[0000-0001-6230-4075]], Xinying Xu[1[0000-0001-5968-5989]], Mifeng Ren[1[0000-0001-5062-0521]]

[1] Taiyuan University of Technology, Taiyuan Shanxi 030024,China
taolan_1983@126.com

Abstract. Environment perception in satellite signal denied situations is one of the challenging problems for multi-robot collaboration. Visual collaborative mapping is an effective solution. However, visual collaborative mapping using Unmanned Aerial Vehicle (UAV) and Unmanned Ground Vehicle (UGV) encounters difficulties such as heterogeneous map fusion and high memory demand for saving the fused map. This paper proposes a heterogeneous map fusion method for Generalized Voronoi Diagram (GVD) and octree map (OctoMap)using point cloud as the transitional map. The UAV builds a global GVD map of the reconnaissance. The UGV establishs OctoMap of the local area after moving to the mission area according to the guidence provided by the UAV. Both the global GVD and the local OctoMap are generated according to point cloud generated by ORB-SLAM3 implemented in the UAV and the UGV. The fused map is presented in the form of high-precision OctoMap in the mission area and GVD map in the non-mission area to reduce memory cost. The proposed map fusion method is demonstrated in a simulation environment.

Keywords: Multi-robot collaboration, map building, heterogeneous map fusion, GVD map, OctoMap.

1 Introduction

Multi-robot collaboration between UAVs and UGVs shows great potential for military and civilian applications to perform special tasks[1]. Accurate location and environment information are the premise for navigation and other higher level tasks. However, it is often difficult to obtain usable satellite signals for localization and navigation in forested areas, built-up urban environments, or indoor environments.

Visual collaborative mapping such as collaborative visual simultaneous localization and mapping (CVSLAM), is an effective solution[2] for satellite signal denied environments. Since vision sensors are low cost, lightweight and small, they are easy to integrate into UAVs and UGVs. The UAV is usually used to build a rough global map of the whole environment, while the UGV is often employed to perceive detailed environment information for target area. Thus, CVSLAM usually requires the fusion of heterogeneous maps since the map built by UAVs and UGVs usually involvs the question of different forms or scales. Furthermore, the high memory requirements needed for saving the fused global map is still open to be addressed.

Common forms of maps built by the UAV and the UGV are metric maps and topological maps. For navigation, grid maps are commonly used dense metric maps, but take up large memory and are not suitable for large-scale map building scenarios[3]. Topological maps, on the other hand, are another form of compact map representation, where nodes in the graph correspond to specific locations in the environment and edges represent passages between different nodes, which is suitable for representing large-scale environments [4], but not suitable for accurate navigation since each node may reprsent a large area and each edge is actually a passable passage without fixed width. For UAV-UGV collaboration scenarios, one type of map often fails to meet the application requirements for navigation in large-scale environemt [5][6]. Heterogeneous map fusion technology can fuse different types of maps to obtain a more adaptable map with different navigation resolution[7].

The innovation of this paper is to save memory space for storing maps. We fuse Generalized Voronoi Diagram (GVD) maps with octree map (OctoMap), so that the UAV builds a global GVD map and determines the target area (usually some nodes in GVD); the UGV travels to the target area according to the GVD map and builds a local OcTOMap. The global GVD map and the local OcTOMap are matched through the transitional point cloud map. Then, we get a global consistent map with coarse environment information for non target area and fine details for target area, which reduces the memory requirement for large scale environment mapping.

2 Related work

The map fusion technique is divided into: map alignment and map fusion. Map alignment is the process of finding appropriate spatial coordinate transformations between local maps. Since mapping alignment is beyond the research of this paper, we assume that the map data have been matched and aligned. Map fusion is the process of merging aligned map data to form a complete and consistent map. Heterogeneous map fusion, is a technique that fuses map data with different geometric structures and attribute information.

The fusion approaches to metric and topological maps include hybrid fusion and hierarchical fusion. Hybrid fusion can address the specific shortcomings of one representation with respect to the other.The study in [8][9] requires a global metric map and also a topological map is extracted from the metric map, which results in a heterogeneous fusion of metric and topology. However, this method does not take memory requirements into account, is not applicable to large-scale environments.

The hierarchical fusion method, on the other hand, represents different maps hierarchically, preserves the advantages of different maps, and can satisfy different application requirements. [10][11] developed the Spatial Semantic Hierarchy (SSH), which can deal with incompleteness or uncertainty in the information depending on the form of the particular localization or navigation problem to be solved, maintaining the representation of the environment in the form of a hierarchy of maps.

However, there is a limit to how efficiently a single robot can work. [12] utilizing multiple UGVs, decentralized sampling is performed in unknown areas according to a set sampling plan. Firstly, the global topology map and the grid map are established independently, and then the nodes of the topology map and the key points of the grid map are correlated and fused by using artificial environment property(such as Radio Frequency Identification (RFID) tags) with image features in the raw data of the sensors. Finally, the heterogeneous fusion of grid and topology is completed in the form of hierarchical fusion.

There is no single sensor that can acquire map data. In the study of [13][14], grid maps and topological maps were created using LiDAR. Subsequently, features are extracted from the raw LIDAR scan data, based on this, topological nodes and connectivity are extracted to create grid-topological maps. In [15], multiple ultrasonic sensors are utilized to obtain environmental information and the multi-sensor information is fused using a BP neural network algorithm. Finally, both the grid map and the topological map are created at the same time, both directly obtaining the grid-topological map. However, both of the above approaches do not consider the size of the storage space, which does not meet the memory requirements of large scene mapping. [16] enables a heterogeneous team of robots equipped with radar and cameras to perform mapping, utilizing a hierarchical system to preserve the environment with metric-semantic topological maps, and ultimately employing target-level semantic topological maps for map fusion, which greatly improves the work.

The heterogeneous fusion method of GVD maps and OctoMaps proposed in this paper enables the creation of a globally consistent heterogeneous map of the environment, meeting the requirements for large-scale environment mapping while maintaining low memory costs.

3 Heterogeneous Map Fusion of Generalized Voronoi Diagram (GVD) and OctoMap

This brief presents a heterogeneous fusion method of GVD maps and OctoMaps based on point clouds, which includes three steps: the creation of GVD maps, the creation of OctoMaps, and the fusion of GVD maps and OctoMaps.

3.1 Main idea of Heterogeneous map fusion of GVD maps and OctoMaps

In this paper, a heterogeneous map fusion approach of point cloud-based GVD maps with OctoMaps combines the advantages of topological maps and 3D raster maps to provide more comprehensive and accurate environmental map information. Specifically, the UAV traverses the entire workspace to create a global GVD map, and at the same time determines the mission area in the GVD map. The pilot UGV travels to the mission area according to the route of the GVD map, establishes the OctoMap of the mission area, and fuses this detailed map with the GVD map to obtain a global consistent map of the large scene with coarse details. The final form of the heterogeneous map is demonstrated in Figure 1.

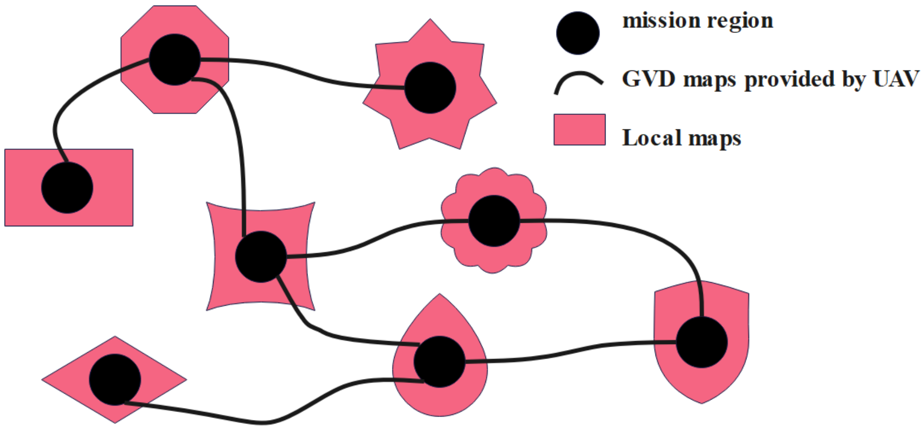


Fig. 1. Pilot UAV provides GVD maps to UGVs, and UGVs travel to the mission area according to the path to build local OctoMaps.

3.2 Creation of the GVD

Voronoi Diagram, also known as Tyson polygon or Dirichlet tessellation, is composed of consecutive polygons formed by the vertical bisectors of the lines connecting two neighboring points. It has a generator within each polygon, and the distance from each polygon to this generator is shorter than the distance to other generators. Points on the boundary to generate this boundary of the generator is equal in distance, and neighboring polygon boundaries to the original neighboring boundaries as a subset, can be expressed in the following formula:

$$\mathrm{V}_i = \{x: \forall j \neq i, d(x,p_i) \leq d(x,p_j)\}, i,j \in \{1,2,\ldots,\mathrm{m}\} \qquad (1)$$

where V_i is the *i-th* polygon, *x* is any point inside or on the edge of the *i-th* polygon, p_i is the generator inside the *i-th* polygon, p_j is the generating element inside the *j-th* polygon neighboring the *j-th* polygon, $d(x,p_i)$ is the Euclidean distance between the point of *x* and the generating element p_i , and $d(x,p_j)$ is the Euclidean distance between the point of *x* and the generating element p_j. $d(x,y)$ is calculated as:

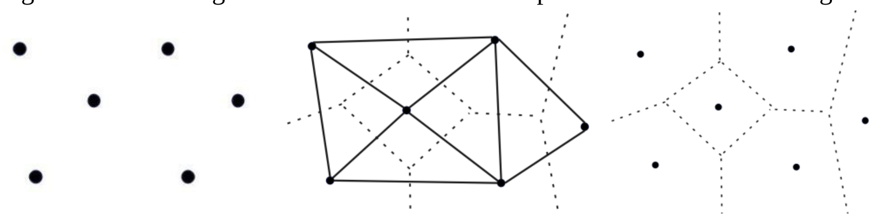
$$d(x,y) = \| x - y \|_2 = \sqrt{\sum_{k=1}^{n}(x_k - y_k)^2} \qquad (2)$$

Compared to Voronoi diagrams, GVD can handle a wider range of objects. For these objects, GVD maps can be divided into different regions in space so that the points in each region are closest to the object represented by that region and farthest from other objects.

In this paper, GVD maps are generated from point cloud data by constructing Delaunay triangles and connecting the centers of the circumcircles of adjacent triangles. The final generation of GVD maps with passable paths also requires the elimination of Voronoi edges intersecting with obstacles. We use ORB-SLAM3, which is a sophisticated and widely used visual SLAM framework, to generate the transitional map—point cloud map. However, the point cloud map usually involves thousands even millions of points to be processed, which may hinder real-time applications. Thus, point cloud sparsification is necessary before further processing.

In this paper, we use radius search with farthest point sampling to sparsify point cloud. That is, select a part of the point cloud in the region, such as corner points, that can represent a subset of points where the distance traveled has changed, as the center point of the query. And specify a search radius to limit the search scope, traversing each point in this radius. First, the distance between the current point and the reference point is calculated to determine whether the distance is within the search radius, and if it is within the search radius, then the point is eliminated. The next reference point is chosen to be approximately the distance from the previous reference point to the search radius continue traversing the set of points in the region until the end to reduce the density and number of points. Then a threshold is set according to the maximum height at which the UGV can pass to remove some of the point cloud data. While because the GVD map is built on a 2D plane, only the first two dimensions of the point cloud data are needed, but the point cloud data is a 3D point set describing the environment. So it is possible to save the first two dimensions in the non-task area first, and wait for the GVD map to be built, and then save only the GVD map to further reduce the data storage and computation overhead.

After processing the point cloud data, the construction of Delaunay triangles begins. After traversing the processed point cloud data, the point sets that can form acute triangles are automatically linked into a triangular mesh, and then the remaining point sets are inserted. That is, for *n* points on the plane, when constructing a triangle net, the requirement that three interior angles of the triangle be acute angles should be met, and at the same time the points and the triangle number should be recorded, listed into the triangle chain table. Figure 2 shows the construction process of the Voronoi diagram.



(a) (b) (c)

Fig. 2. (a) A few processed point cloud data. (b) Point cloud data are randomly connected according to the acute triangle standard, and the perpendicular bisector of each side is made after forming a Delaunay triangle. (c) Voronoi diagram is made by removing the Delaunay triangle.

When inserting the remaining set of points, the triangles whose outer circles contain the insertion points are to be found in the triangle chain list, the common edges of these triangles are to be removed. At the same time the insertion points are to be connected with the vertices of these triangles, completing the insertion of a point into the Delaunay triangle chain list. After that, the vertical bisector is retained and the Delaunay triangles are removed to form a polygonal net with the vertices of each triangle as the generators and the vertical bisector as the edges, which is also known as the Voronoi diagram. Figure 3 shows the insertion of the point P in the Delaunay triangular chain table.

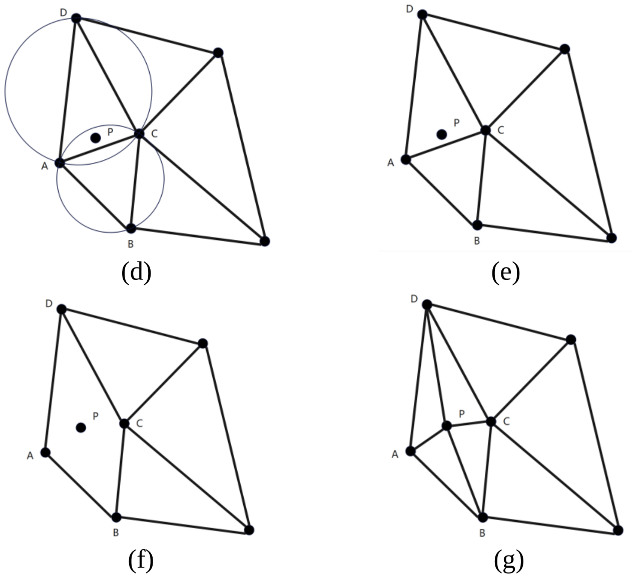


Fig. 3. (d) Insertion of point P after forming Delaunay triangle. (e) The two outer circles containing the P point are found. (f) The common edges contained in the two outer circles are deleted. (g) Point P is connected to complete the insertion of a point in the chain table of Delaunay triangles.

The GVD map with passable paths is built on the Voronoi diagram by connecting all the generators and filtering some Voronoi edges: remove the edges when they are connected to or cross an obstacle boundary, otherwise keep them. Finally we can build a set of passable paths away from all the obstacles. Figure 4 shows an example of a Voronoi diagram and a GVD map.

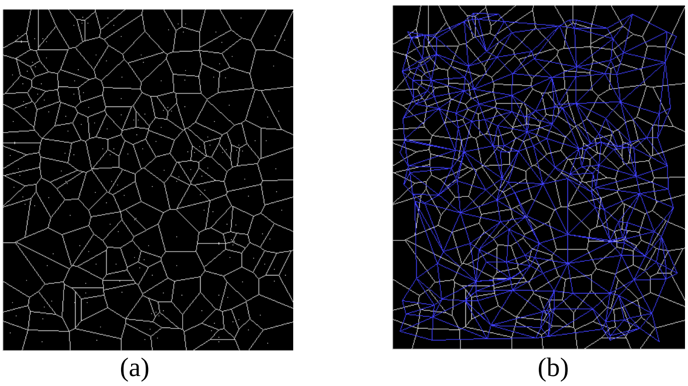


Fig. 4. (a) Example of a Voronoi diagram generated using random points, with white dots as generators and white lines as Voronoi diagrams. (b) Example of a GVD map with all passable paths, with blue lines as passable paths.

3.3 Establishment of OctoMaps

In this paper, we use point cloud data to directly generate OctoMaps (Octomap), which compresses the updated maps with adjustable resolution and saves more memory space by storing the maps in the form of Octomap[17].

Octomap is a spatial partitioning data structure. The cube of the octree before partitioning is known as the root node and after partitioning to maximum resolution is known as the leaf node, also known as the voxel. Each node in Octomap represents a cubic volume which is recursively partitioned into eight sub-volumes until a leaf node is reached. The actual data structure of Octomap it is a tree root that keeps expanding downward, dividing into eight branches at a time, until it reaches the leaves. The leaf nodes represent the highest resolution. For example, if the resolution is 0.01m, then each leaf is a small square of 1cm3. In Octomap, probabilities are needed to express whether a child node of a square is occupied or not. E.g., expressed in terms of a floating-point number $x \in [0,1]$ with an initial value of 0.5, if it is continually observed to be occupied, then let this value will be continually increasing; conversely, if it is continually observed to be blank, then just let it be continually decreasing. However, when a node in the square is all or none occupied, then with a probability of 1 or 0, the node does not need to be expanded, and a map that is blank from the start is then a root node without a complete tree, so that UGV can dynamically model the environmental information in the map.

P is denoted as the probability that a node is occupied or not, belonging to (0,1), and $\alpha \in$ R, denotes the logarithm of the probability. A probability P can be transformed onto the full real number space R by means of the logit transform:

$$\alpha = logit(P) = log(P/(1-P)) \qquad (3)$$

This is an invertible transformation and vice versa there:

$$P = logit^{-1}(\alpha) = \frac{1}{1+\exp(-\alpha)} \qquad (4)$$

Suppose the node is n, t = 1, ... , T moments, the observed data are z_1, ... z_T, and let the logarithmic value of the probability of a node from the beginning to T moments be L (n|$z_{1:T}$), and according to the derivation of the Octomap, the information recorded at the moment T+1 is:

$$L(n \mid z_{1:T}) = L(n \mid z_{1:T-1}) + L(n \mid z_T) \qquad (5)$$

Written in probability form as:

$$P(n|z_{1:T}) = \left[1 + \frac{1-P(n|z_T)}{P(n|z_T)} \frac{1-P(n|z_{1:T-1})}{P(n|z_{1:T-1})} \frac{P(n)}{1-P(n)}\right]^{-1} \qquad (6)$$

In this paper, the target point is an area. When the UGV travels to the area, it directly takes the processed point cloud data, sets the voxel and node thresholds as needed to control the resolution of the Octomap. The first cube is built to the actual prescribed scale of the scene, and the unit elements are put sequentially into cubes that can accommodate and have no child nodes. When the Octomap needs to be updated and optimized, the tree structure can be dynamically adjusted, such as merging the nodes first and then re-splitting the nodes, in order to maintain the real-time and accuracy of the map.

3.4 GVD map and OctoMap map fusion process

Output the global map built by UAV as a GVD map, output the local sub-map built by UGV as an OctoMap. Using the topology map as the underlying map, the OctoMap and the global topology map are fused using the partitioned hierarchical map fusion method.

When the local OctoMap is fused with the global GVD map, different fusion algorithms are executed according to whether the current region is a target region or not. If the current region is not the target region, the edges in the global GVD map are corrected according to the actual running path of UGV; if the current region is the target region, the local OctoMap is fused with the node regions in the global GVD map to obtain the OctoMap of the target region. This partitioned hierarchical map fusion method can not only meet the needs of navigation-based applications but also reduce the storage density of the map in the non-target region, which can realize the compressed storage of large-scale maps. The specific process is as follows.

Firstly, the UAV utilizes the ORB-SLAM3 algorithm to obtain a global sparse point cloud. In order to meet the demand of real-time application, the point cloud is filtered and sparsified, and then the global GVD map is built and the mission area is marked. The pilot UGV travels according to the route of the GVD map, and on the way, the UGV needs to turn on the obstacle avoidance element to prevent it from encountering dangers on the way. When the UGV travels to the mission area, it will use the ORB-SLAM3 algorithm to explore the nearby environmental information and generate an OctoMap. At this time, the UAV stays in the air, updating the global GVD map at all times, waiting for the UGV to finish building the OctoMap of the target area. After the map is built, the UGV continues to follow the GVD map to the next target area to repeat the previous task. Figure 5 shows the map fusion process.

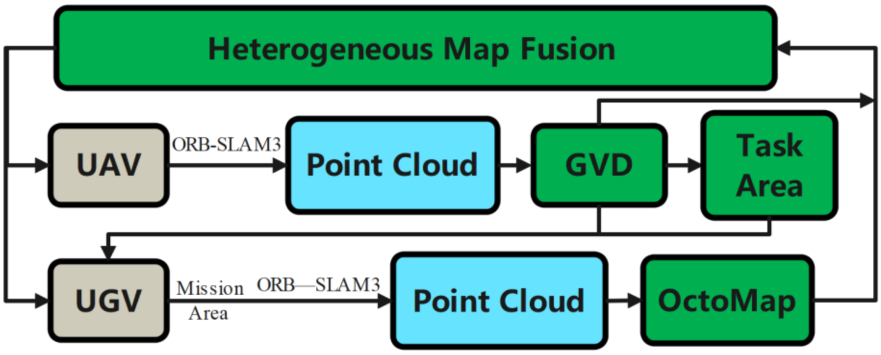


Fig. 6. The UAV generates a GVD map through the point cloud data and marks out the mission area. The UGV drives to the mission area through the GVD map and generates an OctoMap. Finally generates a globally consistent map that integrates the GVD map and the OctoMap.

4 Experiments

We build a simulation environment in gazebo platform, shown in Fig.6. We use the keyboard to control UGV to traverse around in the environment, simulating the UAV in the real environment. ORB-SLAM3 is adopted to collect the environmental point cloud data, simulating the UAV perceiving the global information, so as to build out the GVD map. The the robot is run again, moving from a random positon to the target area guided by the global GVD map, simulating the UGV traversing to the target area to the established the local OctoMap.

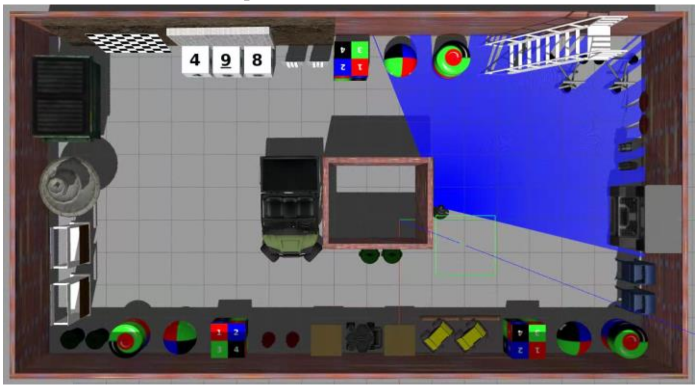


Fig. 6. Live view of the simulating environment.

Since the point cloud acquired by UGV is dense, the computational and storage costs are high, resulting in a slow process of creating Voronoi diagrams. Therefore, it is necessary to perform point cloud sparsification.Each edge of a polygonal obstacle is simulated with a small line segment composed of a sequence of processed discrete point sets, and then a Voronoi diagram is constructed directly using the Voronoi construction algorithm using these approximated discrete points as input. Figure 7 shows the Voronoi diagram generated directly using the sparsified point cloud data.

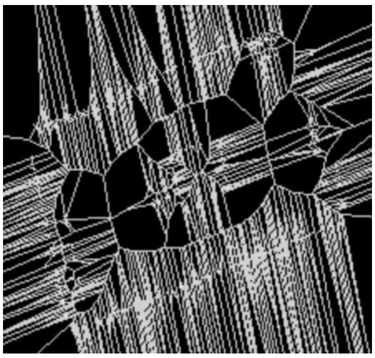


Fig. 7. Voronoi diagram generated directly using the sparsified point cloud data.

In the next step, the edges of the obstacles in the environment are detected by finding the vertices of the obstacles, in setting up a range of spacing between the detected vertices and the obstacles, eliminating the Voronoi edges whose vertices are included in the range, and the rest of the Voronoi edges constitute the set of drivable paths as far as possible from all the obstacles to form the final GVD map. Figure 8 shows the final GVD map.

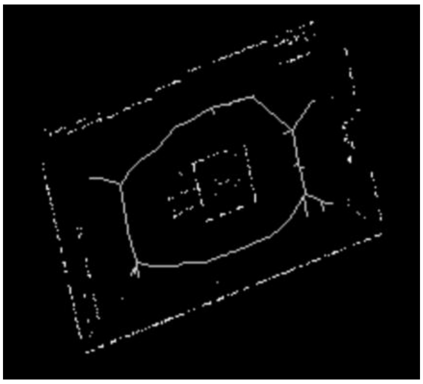


Fig. 8. GVD Map.

The UGV starts the creation of the OctoMap by overall path planning, traveling along the passable path on the GVD map, and then traveling to the mission area by the local shortest path. Firstly, an OctoMap centered on the target point is determined, and then the processed point cloud data is used to directly generate the OctoMap. Figure 9 shows an example of the local OctoMap created when the UGV moves to the lower left corner, the red circle is the starting point, the red line is the real travel route, the blue star is the mission area, and the arrow point to an OctoMap of the target area.

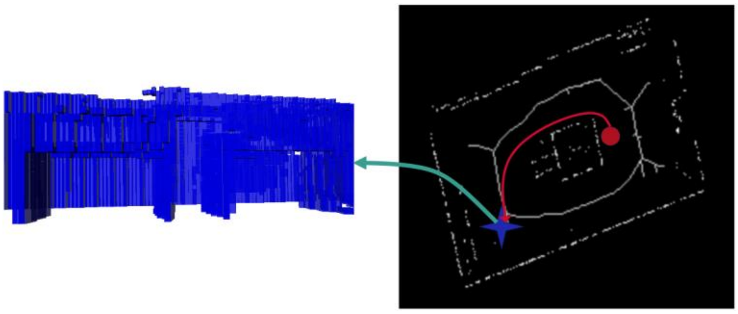


Fig. 9. The UGV travels to the blue star area (mission area) from the red circle (starting point), creating a local Octomap.

The resulting GVD map built by the UAV is used as the base map for navigation, and the OctoMap built by the UGV provided at the mission area is completed.

Checking on the memory used to save maps, we can see that the global map without fusing GVD map and OctoMap takes up 106.51KB, where the global GVD map requires 6.71KB of memory, and the global OctoMap requires 99.8KB. While global map using the proposed map takes up only 12.96KB, where the global GVD map requires 6.26KB of memory, and the local OctoMap built with the task area requires 6.7KB. The proposed method result in 93.55KB memory decrease when only one target area is considered.

The method of generating GVD maps and OctoMaps using occupancy grid maps requires 164KB of memory to store the occupancy grid map. The established global topological map takes 11.24KB, and the local OctoMap for the task area requires 13.8KB, totaling 25.04KB. In comparison, the proposed method requires an additional 12.08KB, which is twice the memory usage of the proposed method. This clearly demonstrates that the method presented in this paper effectively reduces memory usage.

5 Conclusion

In this paper, a heterogeneous map fusion method of GVD map and OctoMap based on sparse point cloud is explored for the collaborative map building work of multiple robots in satellite signal rejection environment. The advantage of UAV's wide view is utilized to build a global GVD map. The global GVD map is provided to the UGV, so that the UGV can navigate itself while staying away from the obstacles in the process of traveling, which ensures the safety of traveling. When the UGV reaches the target point, the UGV builds a local OctoMap, which improves the accuracy of the map and at the same time reduces the occupancy rate of the memory. However, we assume the global map built by UAV and the local map built is aligned before fusing, which is not always true in reality. Future work will loose this assumption to comply with more realistic setting.

Acknowledgments.

This research was funded by the National Natural Science Foundation China, grant number 62073232, Foundation for Scientific Cooperation and Exchanges of Shanxi Province, grand number 202104041101030, and the Natural Science Foundation of Shanxi Province, grant number 201901D211079.

References

1. Li,X.,Yin,X.,Li,S.: Cooperative Event Triggered Control for Multi-Robot Systems with Collision Avoidance. In: 2021 40th Chinese Control Conference (CCC), pp. 5460-5465.IEEE.Shanghai,China (2021)
2. Nemra,A.,Aouf,N.:Adaptive decentralised cooperative vision based simultaneous localization and mapping for multiple UAV. In: 2011 19th Mediterranean Conference on Control & Automation (MED), pp. 546-551.IEEE.Corfu,Greece (2011)
3. Gao Xiang, Zhang Tao.: Fourteen Lectures on Visual SLAM-From Theory to Practice. 2nd edn. Beijing: Electronic Industry Press (2019)
4. Saeedi,Sajad, Paull, Liam, Trentini, Michael.: Group mapping: a topological approach to map merging for multiple robots. IEEE Robotics & Automation Magazine 21(2), 60-72 (2014)

5. Saeedi,S.,Trentini, M.,Seto, M.: Multiple-Robot Simultaneous Localization and Mapping: A Review. Journal of Field Robotics 33(1), 3-46 (2016)
6. Andersone,I.: Heterogeneous map merging: state of the art. Robotics 8(3), 74 (2019)
7. Yue,Y F.,Senarathne,P N.,Yang,C L.: Hierarchical probabilistic fusion framework for matching and merging of 3-D occupancy maps. IEEE Sensors Journal 18(21), 8933-8949 (2018)
8. Song,K.Liu,W.Chen,G.Xu,X.Xiong,Z.:FHT-Map: Feature-Based Hybrid Topological Map for Relocalization and Path Planning. IEEE Robotics and Automation Letters 9(6), 5401-5408, (2024)
9. Vaščák,Ján,Dušan Herich.: Map Merging for Multi-Robotic Applications. In: 2023 IEEE 21st World Symposium on Applied Machine Intelligence and Informatics (SAMI), pp. 21-26. IEEE, Herl'any, Slovakia (2023)
10. The Spatial Semantic Hierarchy, http://www.cs.utexas.edu/users/qr/papers/Kuipers-aij-00.html, last accessed 2024/6/25
11. Benjamin Kuipers,Joseph Modayil,Patrick Beeson,Matt MacMahon,Francesco Savelli.: Local Metrical and Global Topological Maps in the Hybrid Spatial Semantic Hierarchy. In:IEEE International Conference on Robotics and Automation (ICRA 2004), pp. 1050-4729. IEEE, New Orleans, LA, USA (2004)
12. Ferreira,João Filipe.: T-SLAM: Registering topological and geometric maps for robot localization in large environments. In: 2008 IEEE International Conference on Multisensor Fusion and Integration for Intelligent Systems, pp. 392-398. IEEE, Seoul, Korea (South) (2008)
13. Research and implementation of lidar-based grid-topological map construction algorithm, https://d.wanfangdata.com.cn/thesis/D01716176, last accessed 2024/6/25
14. Research and Implementation of Grid-Topology Map Construction Algorithm Based on Lidar,https://d.wanfangdata.com.cn/thesis/D01716176, last accessed 2024/6/25
15. Xu Bing,Xiao Nanfeng.: A hybrid grid-topology map construction algorithm for robot navigation. In: National Advanced Manufacturing and Robotics Summit, pp. 6-11. The Third National Advanced Manufacturing and Robotics Summit Forum, Chengdu,China (2007)
16. Xu Liu,Jiuzhou Lei,Ankit Prabhu,Yuezhan Tao,Igor Spasojevic,Pratik Chaudhari, Nikolay Atanasov, Vijay Kumar.:SlideSLAM: Sparse, Lightweight, Decentralized Metric-Semantic SLAM for Multi-Robot Navigation. arxiv:2406.17249 (2024)
17. Hornung,Armin.: OctoMap: an efficient probabilistic 3D mapping framework based on octrees. Autonomous Robots 34, 189 - 206 (2013)