Collaborative Mapping with IoE-based Heterogeneous Vehicles for Enhanced Situational Awareness

Jorge Peña Queralta[1], Tuan Nguyen Gia[1], Hannu Tenhunen[2], Tomi Westerlund[1]
[1]Department of Future Technologies, University of Turku, Turku, Finland
[2] Department of Electronics, KTH Royal Institute of Technologies, Sweden
Email: {jopequ, tunggi, tovewe}@utu.fi, hannu@kth.se

***Abstract*—The development of autonomous vehicles or advanced driving assistance platforms has had a great leap forward getting closer to human daily life over the last decade. Nevertheless, it is still challenging to achieve an efficient and fully autonomous vehicle or driving assistance platform due to many strict requirements and complex situations or unknown environments. One of the main remaining challenges is a robust situational awareness in autonomous vehicles in unknown environments. An autonomous system with a poor situation awareness due to low quantity or quality of data may directly or indirectly cause serious consequences. For instance, a person's life might be at risk due to a delay caused by a long or incorrect path planning of an autonomous ambulance. Internet of Everything (IoE) is currently becoming a prominent technology for many applications such as automation. In this paper, we propose an IoE-based architecture consisting of a heterogeneous team of cars and drones for enhancing situational awareness in autonomous cars, especially when dealing with critical cases of natural disasters. In particular, we show how an autonomous car can plan in advance the possible paths to a given destination, and send orders to other vehicles. These, in turn, perform terrain reconnaissance for avoiding obstacles and dealing with difficult situations. Together with a map merging algorithm deployed into the team, the proposed architecture can help to save traveling distance and time significantly in case of complex scenarios.**

***Index Terms*—swarm robotics, heterogeneous swarms, cooperative mapping, Internet-of-Everything (IoE), situational awareness**

I. INTRODUCTION

The development of autonomous vehicles, such as self-driving cars, has had a significant improvement over the last decade. However, it is challenging to achieve fully autonomous vehicles due to a variety of open problems related to technical, ethical, cyber-security, and legal issues [1–5]. Autonomous vehicles need to fulfill strict requirements of reliability and be efficient in terms of energy consumption, path planning, or obstacle avoidance. In addition, autonomous vehicles must be able to achieve high levels of situational awareness. In some critical cases related to natural disasters such as sinkholes or earthquakes, serious consequences may occur due to a delay caused by the inefficiency of path planning and lack of situational awareness. For instance, endangered citizens cannot be saved on time because an emergency vehicle cannot reach a destination. Although many autonomous vehicle systems offer some levels of situational awareness, they fail to model the complexity of the infrastructure (i.e., road blocked due to fallen trees or collapsed houses) during or after the natural disasters. These situations might also occur in rescue missions in remote areas where detailed maps are not available, or exploration missions in underdeveloped countries, including those for delivery of humanitarian aid.

Internet of Everything (IoE) can be defined as a virtual platform where virtual objects, actual objects, human, data and processes can be interconnected and communicate with each other. IoE can be considered as an expansion of Internet-of-Things (IoT) where several advanced technologies such as compressed sensing, mesh wireless communication and hybrid cloud/fog computing architectures are involved. With an increasingly ubiquitous IoE, connected vehicles are becoming a closer reality, as an essential part of fully autonomous operation realization [6, 7]. Therefore, it is expected that autonomous vehicles will no longer rely only on data from on-board sensors for local mapping, localization, and operation. A large network of interconnected vehicles will provide enhanced situational awareness and more accurate and efficient mapping. A better understanding of the environment is crucial for the improvement of autonomous operation technology [1, 8, 9].

In order to provide situational awareness for autonomous vehicles, a combination of IoE and a swarm of vehicles is applied. Particularly, we propose a new architecture for cooperative mapping in an unknown environment with a target destination by a group of heterogeneous vehicles. We focus on scenarios where the main vehicle has been given an objective destination and uses a heterogeneous team of support vehicles to gain an enhanced situational awareness via map merging. We give examples of a real scenario in which the support units significantly influence the global path planning.

The main contribution of this work is to provide a proof of concept for the coordination of collaborative mapping within an autonomous team of heterogeneous aerial and land robots. We propose an IoE-based architecture for task assignment and coordination for local map merging. We put a focus on mapping of unknown areas, with a potential application in search-and-rescue missions in post-disaster scenarios. The architecture we propose is meant for optimizing path planning and minimizing traveling time and distance towards a known

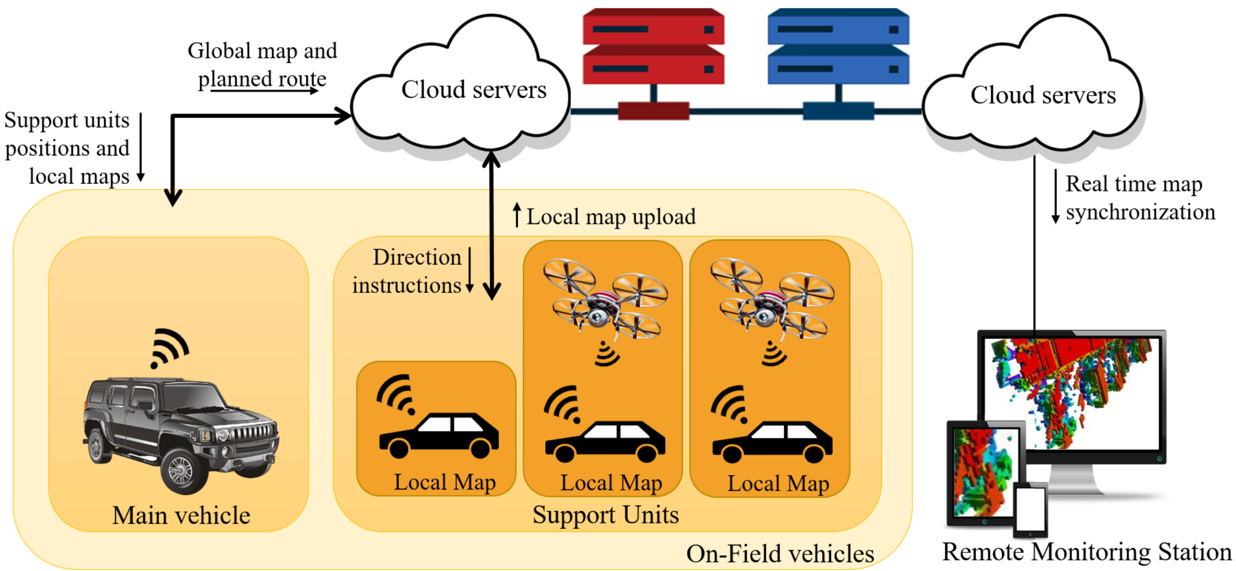


Fig. 1. System architecture

destination. The exploration work is meant to be carried out in an open environment, thus we assume that GNSS-based positioning is available for all vehicles in the reconnaissance mission. If GPS is not available during short periods of time, then inertial measurement units (IMU) and other information such as vision odometry or lidar odometry will be used to estimate the position based on the last known global position.

The paper is organized as follows: Section II presents related work. Section III introduces an IoE-based architecture for enhancing situational awareness. Section IV discusses the implementation and experimental results. Section V concludes the work.

II. Related work

Multi-vehicle localization and mapping was first proposed in 2002 by Williams *et al.* [10]. The authors propose a novel methodology for fusing local maps from multiple agents onto a shared global map. In particular, they simulated a generalization of a Constrained Local Submap Filter to a multi-agent system. The authors also consider the addition of new agents after the mapping has already started and provide a solution for estimating the relationship between the relative frames of reference.

Li *et al.* demonstrate in [11] the benefits that cooperative mapping can offer in challenging environments in comparison with a single robot. They propose a procedure for merging occupancy grid maps in outdoor environments. Their approach includes both estimating indirect relative positions of different vehicles, and a merging function based on occupancy likelihood. The authors demonstrate the utility of their proposed architecture in challenging scenarios where two autonomous vehicles might be too close to each other and therefore partially blocked their vision. Map merging techniques can be applied in such a scenario to enhance the situational awareness of both vehicles and improve their path planning and obstacle avoidance capabilities.

When taking into account the application in a real scenario of a collaborative mapping algorithm, one must take into account the amount of data that needs to be transferred and the bandwidth of the network that is used for coordinating a team of multiple robots and merging their local maps in real time. In particular, if the deployment of such team occurs in a post-disaster scenario, networking infrastructure might be damaged and only mobile networks with lower bandwidth are available. Mostofi *et al.* has demonstrated the efficiency of a collaborative mapping algorithm for a team of UAVs using compressed sensing to reduce the amount of data to be transferred among the agents [12].

Cooperative mapping of unknown environments by a team of heterogeneous robots has already been proposed. For instance, Masehian *et al.* recently present a solution for the coordination and assignment of tasks within the cooperation of multiple robots with the objective of completing a map of an unknown environment [13]. They deploy a team of land robots with different sensor capabilities. The authors specifically focus on the assignment of different tasks to optimize the amount of information gathered by different robots as a function of their sensor capabilities. Also, they introduce an enhanced line merging methodology with fuzzy membership functions for the different robots in order to properly decide on the mergeability of lines from different maps.

As mentioned in Section I, one of the potential application areas for collaborative mapping of a heterogeneous team of robots is a post-disaster scenario. An earthquake, a typhoon or a tsunami can cause a transformation of a well predefined or established map into an unknown environment. Moreover, it is often unsafe for human such as inspectors or lifeguards to explore the damaged area soon after the disaster

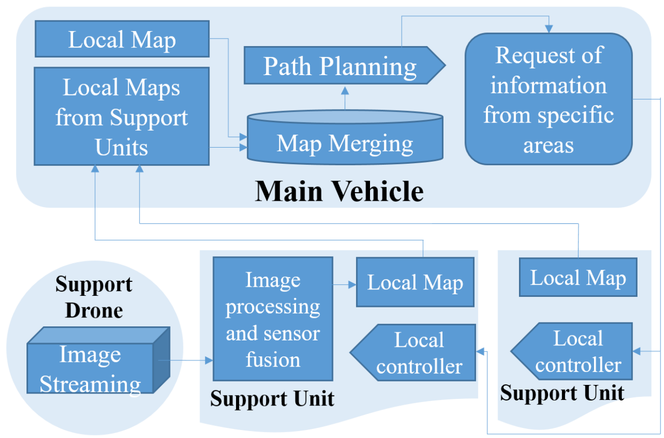


Fig. 2. System Operation

and before the damage has been properly evaluated. This is so because of the potentially unstable structures that might remain in the area. Michael *et al.* carried out an experiment to collaboratively create a 3D map of a compromised building after an earthquake in Sendai, Japan, that was affected by the 2011 Tohoku earthquake [14]. Their experiment produced several 3D voxel grid-based maps of the top three floors of the building. However, the authors discuss that the maps obtained from the scans might be too coarse to be used in a real search-and-rescue operation. Nonetheless, high-quality 3D maps can be achieved with more advanced sensors and improved sensor fusion. While the vehicles that they used in the experiment where tele-operated and not autonomous, the authors stressed how autonomous operation would improve the outcome of their mission.

III. Architecture

We propose an IoE-based system architecture for enhancing situational awareness in autonomous vehicles. The architecture shown in Fig. 2 consists of a swarm of heterogeneous support units, the main vehicle, cloud servers, and an end-user terminal.

The main decisions are taken by the main vehicle, which is given a target position to travel to. In order to optimize the path planning and reduce the traveling time and distance, the main vehicle surveys the position of potential support units and send initial commands to those that caight be near the expected route. The support units can be also other common vehicles (e.g., cars) or special support units (e.g., robots, cars, and drones). The support units gather data and generate local maps which are sent back to the main vehicles for map merging. Depending on the applications, special support units can be deployed at specific points or sent to the target area to check for obstacles and free paths. Some of the special support units can have UAVs in order to survey and collect information of large areas. The collected data from on-board sensors of the support units and drones is processed at the special support units in order to generate a geographical local map. Due to the limited battery capacity, UAVs controlled by the special support units only fly when needed. After each mission, the battery of UAVs is charged with the assistance of the special support units. UAVs can be also deployed from the main vehicles but it is not necessary as UAVs cannot fly over the entire course of the main vehicle's mission and are more efficient at key points during short reconnaissance missions.

The support units are interconnected with each other and connected to cloud services via the 4G/5G wireless protocol. Processed data (i.e., a local map) from all vehicles including main and support units will be transmitted to cloud servers for storing and further processing. An end-user such as system administrators or a driver can access the map with real-time positions of the main vehicles via a browser.

We put the focus on the system-level architecture and definition of commands and data flow. However, the architecture assumes that the positioning of all units with respect to a global coordinate system, such as GPS, is known when the data is shared. In Section V, we use several ground vehicles equipped with GNSS sensors, Lidars and IMUs to test different parts of the proposed architecture.

IV. Path planning and map merging algorithms

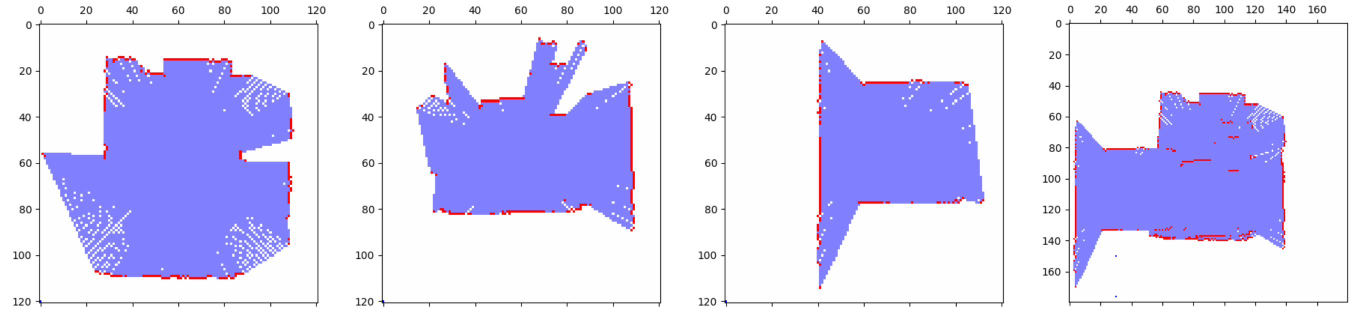
In this section, we introduce the algorithms for path planning, coordination of support units and map merging. The proposed mapping algorithm achieves the best results when some a priori information of the objective environment is available in advance, so that support units can perform reconnaissance in predetermined areas and therefore minimize the time for mapping. This relates to those applications in which a general map of the are is given, but details are unkown.

Due to the constraint computational resources of small UAVs, while the operation is autonomous, the drone path planning is performed on the special support land units which hosts the drone. Moreover, data analysis and compression, and sensor fusion algorithms are implemented on the drone hosts.

A. Mapping and merging local maps

One of the key aspects of the proposed architecture is merging of local maps from the main vehicle and different support units into a single global map. In order to achieve the target, it is required that all vehicles are interconnected and communicate with each other, and connected to remote cloud servers if remote monitoring or control is necessary. Accordingly, they share their positions (i.e. data from GPS) referenced in a global coordinate system. In case these vehicles cannot connect to the Internet, they have to be connected to the same local network, such as a Bluetooth 5 or LoRa mesh network. A position in a global reference may be estimated via simultaneous localization and mapping (SLAM) algorithms, IMU integration, odometry or other methods if GNSS sensors or other similar methods are not available. However, in all those cases, the initial position of the vehicle must be known.

Each of the special support units gathers data about its environment using on-board sensors. Obstacles are detected and stored in a grid occupancy map. We take into account



(a) Local map 1 (a) Local map 2 (a) Local map 3 (a) Merged map

Fig. 3. Real-time map merging in an indoors environment with a known relative positioning. The first three graphs (a-c) show a $12m \times 12m$ map with each cell representing $1/100m^2$. The last graph (d) shows the merged map. The units in all four maps are $1/10m$.

divergent measurements in consecutive mappings by using a value in the interval $[0, 1]$ to represent each cell in the grid. The cell value is increased a fixed value $\delta > 0$ whenever an obstacle is detected in that cell, and decreased a fixed value $\varepsilon < \delta$ whenever it is detected as free. When merging maps from support units into the main vehicle's global map, the main vehicle's map values are given preference with respect to new values. This is so because of the bigger inherent error of the support unit's map due to transmission latency and error in the estimation of its relative position. New values are assumed true for unknown cells, and cells with a higher value in the new local map are given the average value.

Figure 3 shows an example of map merging in an indoors environment with a known relative position between the main vehicle and a single support unit. The last graph (d) illustrates how the main vehicle can have a much more clear understanding of its environment after merging local maps from support units.

B. Path planning and task allocation

The mission starts when the main vehicle is given an objective position to travel to. There might be details that are already known about the area between the initial and objective positions. For instance, if the mission is to be carried out after an earthquake, then hills or forests have probably been less affected than roads due to their size. Therefore, it may be assumed that a path through a known forest will still be impracticable for a search-and-rescue operation. The same applies to major constructions. However, it is unknown for the vehicles whether roads and previous paths are still accessible or have been destroyed or blocked due to the natural disaster.

In order to perform path planning and assignment of routes for support units, the grid occupancy map is converted into an undirected graph. The nodes are generated by grouping together small numbers of free adjacent cells. Then, we find how many connected components are in the subgraphs contained in the different paths between the origin and objective positions. If several paths are found, then the support units are sent in advance for early reconnaissance while the main vehicle dynamically chooses the shortest path. Cells with unknown value are considered to be empty at this step.

Fig. 4. The autonomous cars used in the experiments

Global path planning is achieved by running an adapted Breadth-First Search (BFS) algorithm using the undirected graph generated in the previous step. Then, local path planning with collision avoidance is calculated between the current position and the center position of the next node in the graph. Once the objective positions for the support units are calculated, the instructions are sent over the network (via a direct connection or a cloud server, depending on the deployed network topology). These support units, in turn, can generate instructions for drones that they might be hosting. The raw data obtained from a drone's camera and the support units' sensors is not sent over the network. The raw is analyzed, processed, and compressed into a local map that is transmitted to the main vehicle and Cloud servers for map merging and real-time monitoring, respectively. The raw data can be also stored on the support units so that it can be analyzed after the mission has ended and the algorithms can be optimized.

V. Implementation and Experimental results

In order to assess the efficiency of the proposed system architecture, we compare the distance an autonomous car takes to travel between two points in an unknown environment in different cases. When the traveling distance is longer, it implies that more time is also needed. First, a single vehicle dynamically performs mapping and path planning. In the second experiment, the vehicle performs a mapping with the assistance of support vehicles and a drone.

A. Implementation

The vehicles (e.g., cars) used in the experiments are *1:10 Elektro-Monstertruck "NEW1" BL* models. A radio controller is replaced by Raspberry Pi having WiFi to communicate with other vehicles. The car is controlled via two servo motor control signals, which control the turning angle of the front wheels and the turning speed of all 4 wheels. On top of that, the Raspberry Pi is connected to a RPLiDAR A1M8 LiDAR having a range of 12m and offering a 360 degrees view of the car's environment. The car is equipped with a 9-axis MPU9250 in order to properly capture LiDAR map into an oriented map. In addition, the car has a NEO-M8 GNSS module which can be used to receive data from concurrent reception of up to 3 GNSS (e.g., GPS, Galileo, GLONASS, BeiDou). The GNSS is used for positioning itself as well as calculating the relative positions of support vehicles when local maps referenced in local coordinate systems are merged. Figure 4 shows the cars that are used during the experiments. The LiDAR and GNSS modules are placed outside the car body while battery, Raspberry Pi and MPU9250 are inside.

The drone used in the experiments is made from a DJI f450 drone, a Pixhawk controller, and a Rasberry Pi with a camera. Images collected by the drone's camera are transmitted to the special support unit (i.e., a car). At the support unit, images will be processed by the real-time algorithm for object and distance identification proposed by Ilas *et al.* [15].

For testing purposes only, we connect all cars and drones to the same Wi-Fi Network and place the access point near the starting point in the area that is being subject mapped. In a real scenario, this communication layer would be replaced by a mesh network using a mobile connection such as 4G/5G or another wireless solution with a higher range. During our experiments, we assume that the IP address of the main vehicle is known, while support vehicles can register at any time via a predefined endpoint. In addition to a web server placed at Cloud servers, a web server written in Python using Flask runs on the main vehicle to provide endpoints for receiving the local maps from support units. We also provide a simple monitoring panel to be able to see the evolution of the mapping process in real time. This helps to reduce the latency of transmitting the global map from Cloud to the main vehicle. In this case, a driver of the main vehicles can access the real-time global map with minimum latency. When a system administrator wants to monitor the global map, he/she can access the web server at Cloud server via a browser. Fig. 5 shows the web interface which consists of instant vehicle orientation, LiDAR data, local and global maps, and path planning decisions.

B. Experimental results

All the experiments are carried out in a floor which map is shown in Fig 6. In this floor, there are rooms and furniture. In order to create complex situations which represent chaotic streets caused by natural disasters, several objects are added into the environment. These objects are cardboard boxes and higher than the total height of the car and the LiDAR. In the experiments, vehicles, objects, and LiDAR measurement are scaled down 14 times with respect to their actual size and measurement range. As can be seen a map shown in Fig. 6, in order to reach the destination in the shortest path, an conventional autonomous vehicle follows route 1 → 2 → 3 → 4. At position 4, the car detects that the route is blocked, so it will take the route 5 → 6 → 7. At position 7, the car will go to 8 due to the shorter path to reach the destination point. At position 8, the car detects that the way is blocked; it then follows the route 7 → 9 → 10 → 11 → 12 to the destination. The entire route requires a large amount of traveling time and distance.

When the car is between positions 2 and 3, the path planning algorithm detect two possible paths to the destination. In this experiment, the path to 4 is blocked very soon, but in a real scenario this could be a distance of hundreds to thousands of meters. If there were vehicles nearby those two possible paths, that have previously registered via the known endpoint in the main vehicle, then the latter sends instructions to explore those two areas in advance. The same occurs between points 6 and 7.

An aerial support vehicle (i.e., a drone) can significantly contribute since it is able to see beyond relatively short obstacles that completely block a view of ground vehicles. For instance, a drone flying tens of meters above a car can see what is behind a long wall, or a wrecked building after a natural disaster. Due to this extended vision from above, a drone might easily detect additional obstacles behind the nearest one. This can save time for the ground car as it would need to travel around the obstacle in order to see if the path continues behind or is also blocked. This situation is illustrated in the map in Figure 6 in the area between points 2, 3 and 4, and, more clearly, in the area between points 6, 7 and 8. In the latter case, for instance, a drone flying over a car moving between 6and 7 equipped with a camera can detect the blocked path ahead point 8 in advance. Experimental results show that in the same scenario and destination, the traveling distance of an autonomous car in case of without the support of the proposed architecture is 30% more than the case of with the support of the proposed architecture. It can be inferred that 30% traveling time can be saved with the support of the proposed architecture approximately 30% different. The efficiency of the proposed architecture can vary depending on the maps and scenarios.

VI. Conclusion

In this paper, we presented an architecture definition for the coordination of an autonomous team of heterogeneous aerial and land robots that work together on collaborative mapping. Based on the presented concept, we proposed an IoE architecture having heterogeneous support units for enhancing the situational awareness of autonomous vehicles in an unknown environment. In addition, a complete implementation of heterogeneous vehicles including cars and drones using cameras and sensors such as GPS, and LiDAR was carried out. The results show that the proposed architecture helps

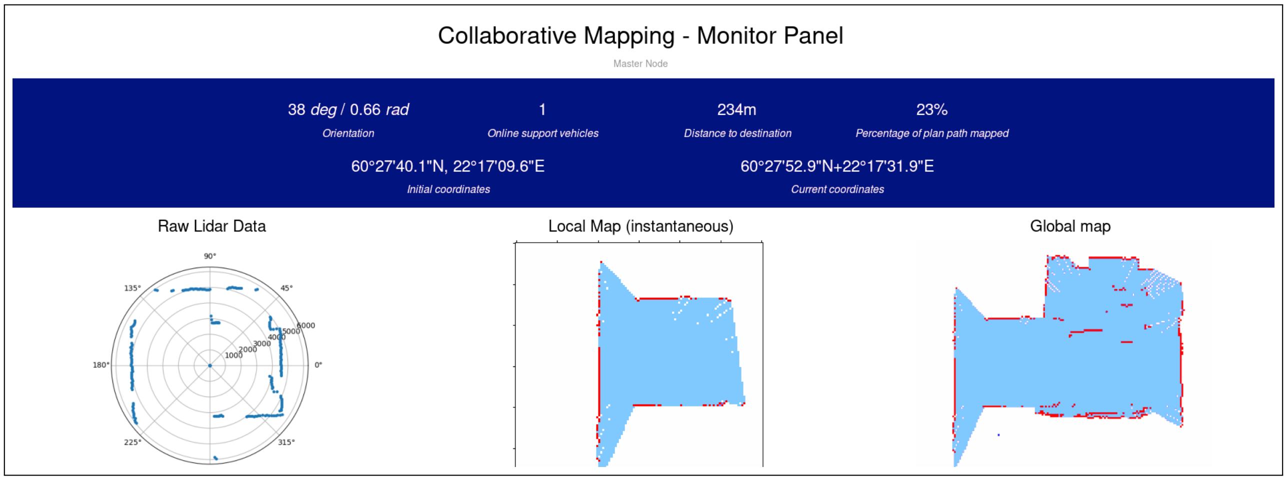


Fig. 5. Web interface with monitoring panel in the main vehicle.

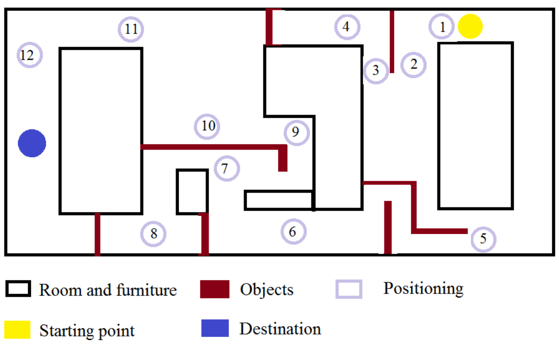


Fig. 6. A map of the experiment environment (Map 1)

to enhance significantly situational awareness. With the large map and the information related to map, an autonomous vehicle can plan in advance for avoiding obstacles and dealing with difficult situations. Furthermore, the results show that the proposed architecture helps to increase the efficiency of an autonomous vehicle by reducing traveling time and distance. In the experiments, 30% traveling distance and time can be saved by deploying the proposed architecture. The efficiency of the proposed can vary depending on the maps and scenarios.

References

[1] R. W. Wolcott *et al.* Visual localization within lidar maps for automated urban driving. In *IEEE IROS 2014*, Sept 2014.

[2] J. Bryson and A. Winfield. Standardizing ethical design for artificial intelligence and autonomous systems. *Computer*, 50(5), May 2017.

[3] N. E. Vellinga. From the testing to the deployment of self-driving cars: Legal challenges to policymakers on the road ahead. *Computer Law & Security Review*, 2017.

[4] J. Bruyne and J. Werbrouck. Merging self-driving cars with the law. *Computer Law & Security Review*, 34(5), 2018.

[5] J. Petit and S. E. Shladover. Potential cyberattacks on automated vehicles. *IEEE Transactions on Intelligent Transportation Systems*, 16(2), April 2015.

[6] E. Uhlemann. Introducing connected vehicles [connected vehicles]. *IEEE Vehicular Technology Magazine*, 10(1), March 2015.

[7] E. Uhlemann. Connected-vehicles applications are emerging [connected vehicles]. *IEEE Vehicular Technology Magazine*, 11(1), March 2016.

[8] O. McAree *et al.* Towards artificial situation awareness by autonomous vehicles. *IFAC-PapersOnLine*, 50(1), 2017. 20th IFAC World Congress.

[9] D. Sirkin *et al.* Toward measurement of situation awareness in autonomous vehicles. In *Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems*, CHI '17. ACM, 2017.

[10] S. B. Williams *et al.* Towards multi-vehicle simultaneous localisation and mapping. In *Proceedings 2002 IEEE International Conference on Robotics and Automation*, volume 3, May 2002.

[11] H. Li *et al.* Multivehicle cooperative local mapping: A methodology based on occupancy grid map merging. *IEEE Transactions on Intelligent Transportation Systems*, 15(5), Oct 2014.

[12] Y. Mostofi *et al.* Compressive cooperative sensing and mapping in mobile networks. *IEEE Transactions on Mobile Computing*, 10(12), Dec 2011.

[13] E. Masehian *et al.* Cooperative mapping of unknown environments by multiple heterogeneous mobile robots with limited sensing. *Robotics and Autonomous Systems*, 87, 2017.

[14] N. Michael *et al.* Collaborative mapping of an earthquake damaged building via ground and aerial robots. In *Journal of Field Robotics*, volume 92, 07 2012.

[15] C. Ilas *et al.* Real-time image processing algorithms for object and distances identification in mobile robot trajectory planning. *Journal of Control Engineering and Applied Informatics*, 13(2), 2011.