2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)
October 25-29, 2020, Las Vegas, NV, USA (Virtual)

City-Scale Grid-Topological Hybrid Maps for Autonomous Mobile Robot Navigation in Urban Area

Shun Niijima[1,2], Ryusuke Umeyama[1,2], Yoko Sasaki[2], and Hiroshi Mizoguchi[1,2]

Abstract— **Extensive city navigation remains an unresolved problem for autonomous mobile robots that share space with pedestrians. This paper proposes a configuration for a navigation map that expresses urban structures and an autonomous navigation scheme that uses the configuration. The proposed map configuration is a hybrid structure of multiple 2D grid maps and a topological graph. The occupancy grids for path planning are automatically converted from a given 3D point cloud and publicly available maps. The topological graph enables the connections between the subdivisions of occupancy grids to be managed and are used for route planning. This hybrid configuration can embed various urban structures automatically and is applicable to a wide range of autonomous navigation tasks. We evaluated the map by generating the proposed navigation map in real city and performing path planning using on the hybrid map. Experimental results demonstrated that the hybrid map can reduce the planning time and memory usage compared to the conventional single 2D grid map based path planning.**

I. INTRODUCTION

Various types of navigation maps that represent the environment are used for the autonomous navigation of mobile robots. Two of the most common ones, 3D point cloud maps and grid maps, are used as prior knowledge for autonomous navigation and applied mainly for localization and path planning respectively. Mobile robots incorporate these maps to achieve autonomous navigation in urban areas [1]–[3].

The 3D point cloud maps are a set of data points in space. The 3D point cloud maps are generally produced by 3D scanners and represent the 3D shape of an object. Localization based on a 3D point cloud map achieves high accuracy even if the Global Navigation Satellite System (GNSS) is inaccessible [4], such as indoors or in downtown high-rise areas. The 2D grid maps are utilized for path planning where the basic idea is to represent the environment as a spaced field of binary random variables that represents the presence of an obstacle. 2D grid maps [5] are possibly the best known type of map for robots and have been increasingly utilized in 2D navigation systems. 2D grid map representations are used in path-planning algorithms such as A* [6], rapidly exploring random tree (RRT) [7], and variations thereof. The 2D grid map has been expanded to 3D for dealing with 3D path planning. The most common scheme for the 3D grid map is Octomap which is based on Octree [8].

[1]Department of Mechanical Engineering, Tokyo University of Science,2641 Yamazaki, Noda-shi, Chiba 278-8510, Japan

[2]National Institute of Advanced Industrial Science and Technology, 2-3-26 Aomi, Koto-ku, Tokyo 135-0064, Japan

*This work was supported by JSPS KAKENHI Grant Number 19J22783.

It is difficult to use a large-scale 3D grid map outdoors, as the data volume grows at a much higher rate than the scale of the region. To reduce the memory usage, 3D information is compressed into a multi-layer 2D grid map [9]. The multi-layer 2D grid maps are generated from 3D point clouds, and the drivable information is managed in each layer. These maps can add new information simply by adding layers. For example, we can generate a dynamic obstacles layer from sensor observations to help avoid dynamic obstacles such as pedestrians. Autonomous navigation based on multi-layer 2D grid maps has enabled self-driving in urban areas of about 1 km^2 [10]. The multi-layer structure is effective for driving in an urban environment that contains a variety of information.

Autonomous navigation based on single nationwide 2D grid map is difficult due to the huge computational cost and memory usage required. Some navigation systems apply a hybrid map [11], [12] to alleviate these problems. The hybrid map is composed of 2D grid maps and a topological graph. These approaches manage multiple 2D grid maps with topological graphs and reduce the computational cost by introducing higher level topological path planning. However, designing these topological graphs on the city-scale required significant time and cost.

In this paper, we propose a grid-topological hybrid navigation map that is automatically converted from a given 3D point cloud and publicly available maps. Publicly available maps (such as OpenStreetMap [13]) are free-to-use maps that are maintained throughout the world. Our proposed map structure expands the range of navigation maps thanks to its easy to implementation due to the automatic conversion based on publicly available maps. Moreover, publicly available maps include rich information of urban areas (streets, tracks, railways, land-use, and building information), which is useful for performing autonomous navigation.

The information in publicly available maps has been incorporated for autonomous vehicles in other studies. In rural areas, a new framework that combines sparse topology maps for global navigation and sensor-based perception systems for local navigation [14] has been proposed. In an urban scenario, publicly available maps were introduced for localization, path planning, and autonomous vehicle control in [15]. This approach uses building information for localization and road information for path planning and performed well in real-world experiments at two locations in Hanover, Germany. A low-cost localization system for a car-like robot on streets was proposed in [16], where information on the intersections (critical points) and lanes (road following) from publicly available maps was utilized.

978-1-7281-6211-9/20/$31.00 ©2020 IEEE

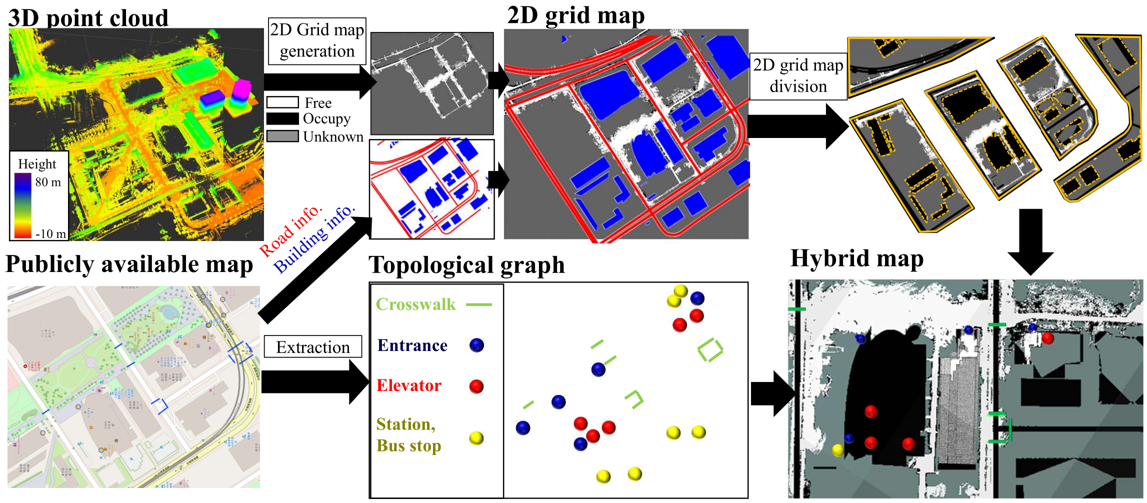


Fig. 1. **Generation process of city-scale grid-topological hybrid navigation map.** Building information, road information and topological graph are extracted from publicly available maps. Building information and road information are integrated into a 2D grid map generated from a 3D map. The 2D grid map is divided based on road information (yellow solid line) and building information (yellow dotted line). The hybrid map is generated by connecting the divided 2D grid maps with a topological graph.

These navigation systems based on publicly available maps have enabled effective support functions for robots that drive on the road.

Our objective in this study is to faciliate large-scale autonomous navigation of mobile robots in a city. The proposed grid-topological hybrid configuration help reduce memory usage and computational costs when performing large-scale path planning. The proposed configuration is used for mobile robots that move not only on the road but also in other scenarios such as moving along a sidewalk, where driving rules do not apply, and hierarchical movements in a building via elevators. The proposed navigation framework is useful for performing autonomous navigation in a city-scale context due to its understanding of the relevant urban structures. Specifically, it is possible to express urban structures, such as hierarchical movements in buildings, that are difficult to express with a single 2D grid map.

This rest of this paper is organized as follows. Section II explains the configuration of the navigation map. Section III describes the proposed navigation framework using a topological graph and 2D grid maps. In section IV, we demonstrate the generation of the proposed hybrid navigation map and evaluate its path planning performance in a real city. We conclude in Section V with a brief summary.

II. City-scale hybrid navigation map configuration

The proposed navigation map is composed of three elements: 3D point clouds, divided 2D grid maps, and a topological graph. An overview of the proposed navigation map is shown in Fig. 1. In this section, we present the details of each of the three elements and explain how the publicly available maps are incorporated into the topological graph for mobile robots in urban areas.

A. Hybrid map configuration

The 3D point cloud map achieves accurate localization even in an urban area where GNSS is not accessible. The 2D grid maps are generated from a 3D point cloud map and contains the information necessary for autonomous navigation, such as drivable areas and speed limit information. The 2D grid maps implement path planning to avoid dynamic obstacles by reflecting these obstacles in real-time from sensor observations. These 2D grid maps are very effective for mobile robots in an urban area with many dynamic obstacles such as pedestrians. In our approach, the 2D grid map adds layers that represent roads, buildings, and crosswalks.

To make the input/output (I/O) with navigation easier, the 2D grid map is divided on the basis of primary road information and building information from publicly available maps. The divided 2D grid maps are then stored as the local 2D grid maps. These divided maps are connected by a topological graph, which is obtained from publicly available maps. The topological graph is composed of nodes and edges. In autonomous navigation, the robot determines the rough route using the topological graph and then locally performs metric navigation by loading a divided 2D grid map. A detailed description of the navigation process is provided in Section III.

B. 2D grid maps

The 2D grid map is generated from a 3D point cloud map. The 3D point cloud maps must overlap with the digital maps so that information from the publicly available maps can be added. Methods for generating 3D point cloud maps based on publicly available maps are being developed to achieve accurate simultaneous localization and mapping (SLAM) outdoors [17]–[21]. Robot can use these methods to generate 3D point cloud maps that overlap with publicly available maps. Such 3D point cloud maps are being rapidly produced around the world, and researchers will be able to utilize them in the near future.

The 2D grid map is used for path planning and motion planning. It is usually generated from a 3D point cloud map using SLAM conducted by the robots themselves [22], [23] and they use the map to plan a route on continuous pavement without obstacles. However, these 2D grid maps generally represent any and all travelable areas for mobile robots, which means the robot might inadvertently perform path planning that moves onto roads exclusively for cars or through private land associated with buildings. To overcome this problem, our 2D grid map adds three layers: road, crosswalk, and building information. In autonomous navigation, path planning is implemented to avoid dangerous places and to enable a mobile robot to move over a crosswalk correctly.

City-scale 2D grid map requires significant memory usage and high computational costs. To make I/O for navigation more manageable, we divide the 2D grid map into local 2D grid maps $\boldsymbol{M} = \{M_1, \cdots, M_n\}$ using primary road information and building information. Large outdoor 2D grid maps are segmented with primary road lines, and indoors, one separate 2D grid map is used. In buildings, one 2D grid map is used for each floor. The divided 2D grid maps are connected using the topological graph (described in C) and are utilized while switching the local 2D grid maps at the time of autonomous navigation.

C. System definition of topological graph

In this subsection, we describe the structure of our topological graph, which is composed of nodes and edges. It is used to connect the divided 2D grid maps and perform simple path planning to the destination.

The nodes define single points, such as the position of elevators and building entrances. A node gives information about point. Each node has three properties defined as $\boldsymbol{N} = \{id, \boldsymbol{p}, s\}$, where:

- id is the unique name of the node, and the node is stored with the divided 2D grid maps,
- $\boldsymbol{p}$ is composed of position $\{x, y, z\}$ and pose $\{q_x, q_y, q_z, q_w\}$ that define the target position and pose of a robot in the world, and
- s is the status information that describes where and what this point is, such as an entrance or an elevator.

The edges define the connections between nodes, such as the space between the start and end points of a crosswalk and the floors where an elevator is able to go. An edge has four properties defined as $\boldsymbol{E} = \{id_1, id_2, c, s\}$.

- id_1, id_2 are the ids of two connected nodes.
- c is the numerical cost for topological path planning. The cost can be set in accordance with the purpose, such as the distance between nodes or the time required to move between nodes. In our work, the cost of an edge is the total cost resulting from path planning on a local 2D grid map, and the edge obtained from publicly available maps in advance, such as a crosswalk or elevator, is defined as a specific cost on the basis of each status. The cost of $Edges$ is defined on the basis of each status.
- s is a description of the status between nodes, e.g., automatic_door, access=yes. It is useful for storing information of additional tags in publicly available maps information. This information is utilized to change behavior during autonomous navigation.

D. Topological graph generation

The topological graph is generated by extracting information from the publicly available maps. The extracted information becomes either a node or edge according to the system definition explained in II.C. Nodes in the same 2D grid map are connected by an edge if it succeeds in that path planning based on the 2D grid map. The extraced edge information defines the connection between the node in different divided 2D grid maps. The proposed method uses the following five types of information:

- **crosswalk:** The crosswalk graph connects the 2D grid maps between local 2D grid maps outdoors. Two nodes and one edge describe each objects. Two nodes of a certain position are set at each end of a crosswalk. The pose of the node is set to the direction of the planned path, that is, the next node. The status of the edges is then defined, e.g., traffic_signals, uncontrolled, or button_operated=yes/no. These description indicate whether is a traffic signal for the pedestrian.
- **entrance:** The entrance graph connects 2D grid maps between the indoors and outdoors. The node of the entrance is defined as one node on the outline of the building. The pose of the node is perpendicular to the building. The status of the edge describes the kind of entrance, e.g., automatic_door, wheelchair=yes.
- **elevator:** The elevator graph connects the 2D grid maps between each floor in the building. The elevator is composed of one node for each floor and edges connect the nodes. The position is set in front of the elevator. The status of the edge is access=yes/no.
- **overpass:** There are areas in which a robot can move in a vertical overlapping direction. Such areas are represented by combining a topological graph and 2D grid maps. An area such as this has a similar expression to the layered structure of a building. The place where the runnable area overlaps vertically is saved in two 2D grid maps, and nodes and edges are generated at each connection.
- **station:** The station nodes are connected by railways and bus networks, which robots use when travelling long distances. The nodes of the station or bus stop set in the landing area. The position is facing the track. The definition of edges is already given by numerous services, such as route search and Google Maps.[1] [2]. These services provide ways to search for routes between stations in various situations.

[1]https://world.jorudan.co.jp/
[2]https://www.google.co.jp/maps/

III. Hybrid map based navigation

The goal of autonomous navigation is to create a route from the current position to the destination on the map. In our navigation framework, to reduce memory usage and the calculation time of path planning on the city-scale, metric navigation is performed while sequentially switching the local 2D grid maps.

A. Navigation system

Figure 2 shows a simple navigation example and the pipeline of the proposed navigation system. Topological path planning selects the start and end nodes. It calculates the plan, that is, the order in which to follow the nodes, from the start to the end node. The module sends the next node as the next local goal to the state manager. After receiving the node and edge status, the state manager determines the next action.

The state manager module sends the navigation task to the metric navigation module. The goal of metric navigation is to reach the position and pose of the received node while switching to a local 2D grid map. This is repeated until the robot reaches the destination.

B. Topological path planning

To determine the start and end nodes, topological path planning module searches for the closest nodes. It defines the node closest to the robot as the start node $\boldsymbol{N_s}$ and the one closest to the destination point as the end node $\boldsymbol{N_e}$. The planned path between $\boldsymbol{N_s}$ and $\boldsymbol{N_g}$ is calculated from topological graphs using Dijkstra's algorithm [24]. The planned path (plan) is a list of nodes from $\boldsymbol{N_s}$ to $\boldsymbol{N_g}$: **Route** $= \{\boldsymbol{N_s} \cdots \boldsymbol{N_n} \cdots \boldsymbol{N_g}\}$.

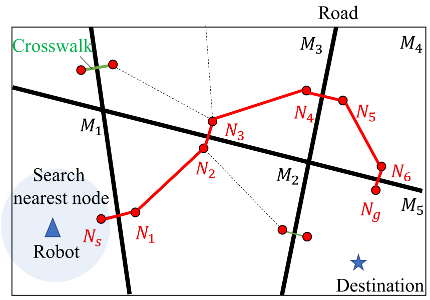
However, the first node $\boldsymbol{N_s}$ may be facing a different direction than that of the destination. To avoid generating a detour path, the module compares the cost of moving from $\boldsymbol{N_s}$ to $\boldsymbol{N_1}$ and that of moving from the current position to $\boldsymbol{N_1}$. If moving directly to $\boldsymbol{N_1}$ has a lower cost, $\boldsymbol{N_1}$ is set as the first destination. The goal node setting is performed in the same way.

C. Metric navigation

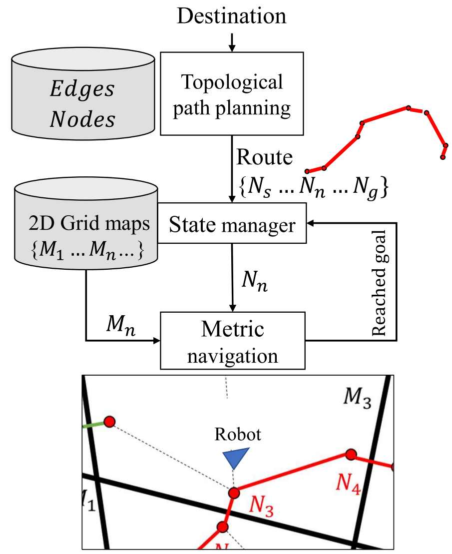
The robot performs metric navigation using the received N_n position and pose as local goals. It does not need to know exactly where a node is in order to avoid wasted motion. When the robot approaches the destination node, it plans a path to the next node. Therefore, the path planning in each 2D grid map needs to be done quickly.

The action of the robot changes depending on the status of the next node and edge. The state manager sends the navigation task in accordance with the status of the topological graph. For example, if the next node and edge is an elevator, the navigation task is to get on the elevator and move next floor. If the next node is on a different 2D grid map, the state manager sequentially loads the 2D grid map containing that node and sends to the metric navigation.

The position of the received node is generated from the navigation map. Path planning may not be possible if there are dynamic obstacles that the navigation map does not show. In such cases, the robot arrives near the destination and receives the next node. When path planning cannot be performed because of a newly installed obstacle, the topological navigation blocks the edge that is obstructed by the obstacle and recalculates the plan.



(a) Simple navigation example

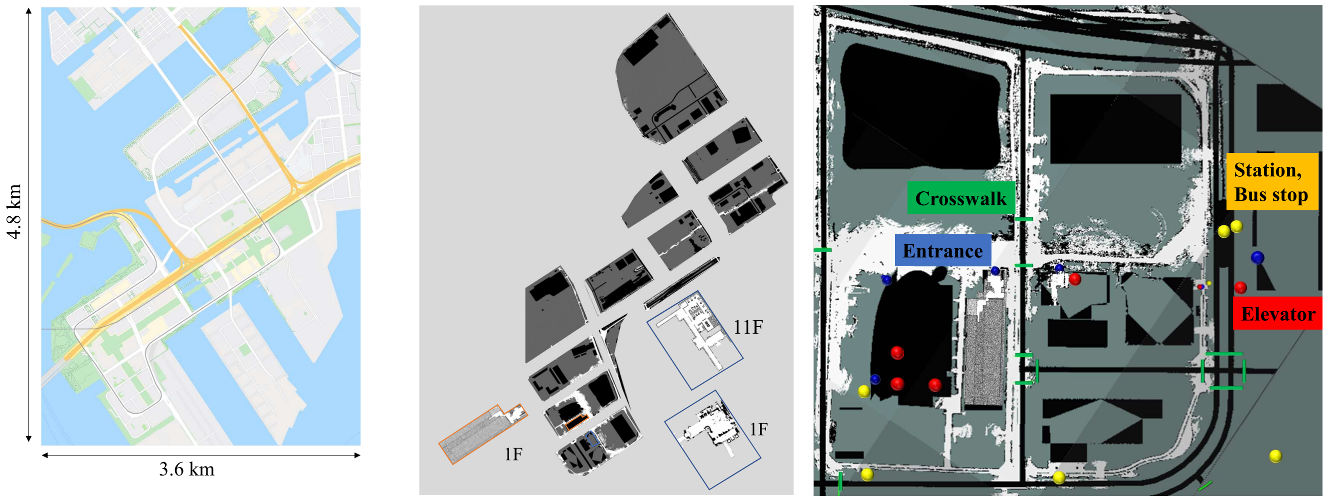


(b) Pipeline of the system

Fig. 2. **Autonomous navigation based on the hybrid map.** (a) shows an example in which six gridmaps divided by roads are connected by a crosswalk topological graph. In (b), the system receives the location of the destination(blue star) and calculates the rough route (red line) by topological path planning. The state manager manages the order of loading the local 2D grid maps and decides which nodes are set to local goals. The metric navigation calculates the path to the node in each local 2D grid map.

IV. Evaluation

This section evaluates the proposed hybrid navigation map by performing map generation for a real city. After briefly describing the specifications of the navigation map, we report the results of the path planning based on the hybrid map. We evaluated the proposed path planning by calculating memory usage and planning time and comparing the results with a conventional method that uses single 2D grid map. The path effectiveness is evaluated by comparing the length with the A* optimal path. To clarify the contributions of the proposed hybrid navigation map, the path planning is performed using



(a) Generated area (b) Divided local 2D grid maps (c) Hybrid navigation map

Fig. 3. **Results of the map generation**: (a)shows the Google Maps view of the generated area. (b) shows the 15 divided local 2D grid maps based on road and building information from OpenStreetMap. Black areas indicate buildings and roads. (c) shows the proposed hybrid map. The color sphere marker represents various nodes, and the green lines represent the edge of the crosswalk.

TABLE I
SPECIFICATION OF THE HYBRID NAVIGATION MAP

Gridmap	
Map size	4.8 × 3.6 km
Resolution	outdoors 0.1 [m] indoors 0.05 [m]
Number of local 2D grid maps	15
Topological graph	
Crosswalks	183
Entrances	31
Elevators	50
Overpasses	2
Stations	9
Total number of nodes	275

two patterns from the path planning results. The results show that our method can perform both large-scale and hierarchical path planning using an elevator which cannot be performed when using the single 2D grid map.

A. Map generation

Two types of 3D point cloud maps are used in the evaluation. One is generated by the Mobile Mapping System (MMS) and the other by Cartographer [25], a state-of-the-art SLAM method. The 3D point cloud maps are corrected by graph-based optimization using building information and some manual correction. The 2D grid map is generated from a 3D point cloud map using a drivable map generation method [9]. The 2D grid map contains three layers: the position of obstacles, speed limit information, and information on whether streets are paved or unpaved. The topological graph is obtained from OpenStreetMap [13], a free, editable map of the whole world built largely from scratch by volunteers and is released with an open-content license. The OpenStreetMap data can be easily obtained and utilized by using the overpass API[3].

The results of the 2D grid map and topological graph generation are shown in Table I. The hybrid navigation map was a 4.8 × 3.6km square in Odaiba, Japan. The resolution of the 2D grid map was different indoors and outdoors to allow for finer path planning indoors. The 2D grid map was divided into 15 local 2D grid maps comprising three indoor and 12 outdoor maps. These navigation maps were basically automatically converted from 3D point cloud maps and OpenStreetMap data.

The generated hybrid navigation map is depicted in Fig.3, where (a) shows the Google Maps view of the generation area, (b)shows the divided local 2D grid maps, and (c) shows an enlarged view of the hybrid navigation map. The hybrid map in the figure includes two outdoor local 2D grid maps. In the 2D grid map, black represents no entry, white represents free, and gray represents unknown. The colored spheres represent the node and the green lines represent the edge of the crosswalk. In the figure, we can see that the roads and building areas have been correctly set as occupied and the local 2D grid maps divided by roads are connected by the edges of the crosswalk.

B. Path planning

We evaluate the path planning by calculating the plan time and memory usage, and comparing the results with a conventional method based on a single 2D grid map. The start and goal positions are determined randomly in the map and path planning is performed over eight trials. We adopted the A* [6] algorithm for path planning in a 2D grid map implemented by robot operation system (ROS) [4].

[3]https://overpass-turbo.eu/
[4]http://ros.org

TABLE II
RESULTS OF TIME AND MEMORY USAGE REQUIRED FOR PATH PLANNING

Trial	Situation	2D grid map		Hybrid maps		
		plan time [s]	memory usage [GB]	plan time (max) [s]	memory usage (max) [GB]	switching 2D grid map no.
1	outdoors	1.3	1.67	0.56 (1.02)	0.19 (0.77)	4
2	outdoors	0.35	0.43	0.10 (0.16)	0.39 (0.42)	2
3	outdoors	0.46	3.44	0.11 (0.37)	1.48 (2.52)	3
4	outdoors	0.31	3.39	0.09 (0.31)	1.52 (2.52)	3
5	outdoors	0.53	2.57	0.06 (0.24)	0.64 (1.02)	5
6	outdoors	-	8.82	0.13 (0.31)	0.80 (2.52)	8
7	indoors, outdoors	-	-	0.01 (0.02)	0.21 (0.42)	5
8	indoors, outdoors	-	-	0.12 (0.43)	0.11 (0.42)	4

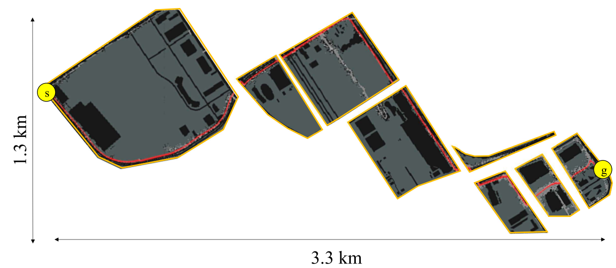
Path planning based on the hybrid map worked correctly for all trials, which indicates that the proposed hybrid navigation map is useful for robot path planning in urban areas. Table II summarizes the results for planning time and memory usage obtained using a single 2D grid map and the proposed hybrid map. The results for the single 2D grid map, where path planning was performed using a 2D grid map with a size including the diagonal line for the start and goal, are shown on the left. The results for the proposed hybrid path planning are shown on the right. Both average and maximum values are shown. The column on the far right indicates the number of 2D grid maps that were switched. Overall, the experimental results demonstrate that the memory usage and planning time of the hybrid map were reduced compared to when using the single 2D grid map.

In trial 6, the start and goal were more than 3 km apart, and the 2D grid map memory size was 8.82 GB. The single 2D grid map-based path planning was not able to perform A* by ROS since the memory usage was too large. Fig. 4(a) shows the results of path planning (red line) based on the hybrid map. Despite the long-distance routes, the path planning was achieved on average with less planning time and memory usage. These results demonstrate that the proposed hybrid navigation is useful for large-scale autonomous movement, which is difficult to achieve with a single 2D grid map. The path planning using the proposed hybrid map does not depend on the distance even in on the city-scale, and it is possible to perform path planning with a fixed memory usage and calculation time. In trials 7 and 8, the scenario involved movement both indoors and outdoors. The path planning could not be calculated with the single 2D grid map since the movement included an elevator in the building. Fig. 4(b) shows the results of path planning in trial 7 based on the hybrid map. Here, we can see that the elevator could be used and the path was planned correctly. This result demonstrates that it is possible to move between such layers.

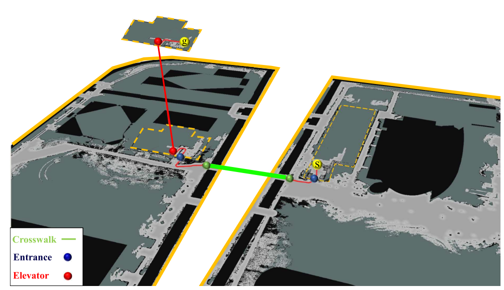
Next, the path effectiveness is evaluated by comparing the path length with that of the A* optimal path. Table III shows the results of path length in each trial. In all trials, the hybrid map-based path planning could calculate almost the same path as the optimal one, while the 2D grid map could not. Fig. 5(a) shows the results of path planning in trial 1. The path based on the hybrid map was a little longer than the one using a single 2D grid map, as the path was drawn through a crosswalk or another graph. These results demonstrate that the hybrid map-based path planning reduces the planning time and memory usage compared to planning using one 2D grid map, and has almost the same path-planning performance.

TABLE III
COMPARISON WITH A* PATH LENGTH.

Trial	Path length [m]	
	2D grid map	Hybrid maps
1	1492.3	1492.8
2	432.6	434.4
3	2247.5	2256.3
4	1807.1	1809.8
5	1985.6	1989.5
6	–	4771.5
7	–	125.4
8	–	1331.9



(a) Large-scale outdoor path planning in trial 6



(b) Hierarchical path planning with elevator in trial 7

Fig. 4. Path planning achieved with hybrid map (could not be achieved with single 2D grid map).

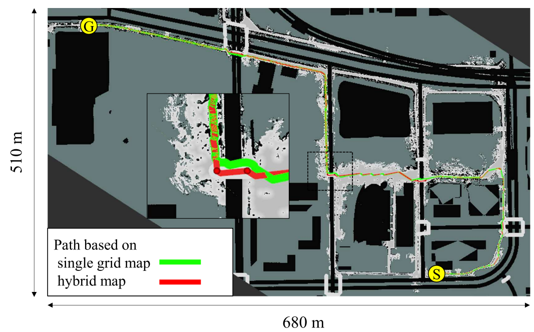


Fig. 5. Comparison of the path lengths in trial 1.

V. Conclusion

In this paper, we have proposed a configuration for a hybrid navigation map that combines 2D grid maps and a topological graph. The hybrid navigation map is automatically generated by given 3D point clouds and publicly available maps, thus enabling city-scale maps to be generated at low cost. We performed experiments in which we generated maps for a real city in Japan and found that the proposed hybrid navigation maps reduced the memory usage and calculation time for path planning. We also found that we could perform large-scale path planning and path planning between tiers, which is not possible using the existing single 2D grid map. These results demonstrate that the proposed hybrid map is effective for a wide range of autonomous navigation tasks in urban areas.

The proposed navigation map can be expanded because information can easily be added to the 2D grid map by adding layers. For example, a moving object layer could be used to execute route planning that avoids places with many people. The new topological graphs from OSM information can be added arbitrarily. Robots may also be able to perform navigation that supports train or bus movements with the addition of a railway or bus graph.

References

[1] R. Kümmerle, M. Ruhnke, B. Steder, C. Stachniss, and W. Burgard, "A navigation system for robots operating in crowded urban environments," in *2013 IEEE International Conference on Robotics and Automation (ICRA2013)*, Singapore, 2013, pp. 3225–3232.

[2] E. Trulls, A. Corominas Murtra, J. Pérez-Ibarz, G. Ferrer, D. Vasquez, J. M. Mirats-Tur, and A. Sanfeliu, "Autonomous navigation for mobile service robots in urban pedestrian environments," *Journal of Field Robotics*, vol. 28, no. 3, pp. 329–354, Mar. 2011.

[3] Q. Yang, D. Qu, F. Xu, F. Zou, G. He, and M. Sun, "Mobile robot motion control and autonomous navigation in gps-denied outdoor environments using 3D laser scanning," *Assembly Automation*, vol. 39, no. 3, pp. 469–478, Aug. 2019.

[4] A. Ohsato, Y. Sasaki, and H. Mizoguchi, "Real-time 6Dof localization for a mobile robot using pre-computed 3D laser likelihood field," in *2015 IEEE International Conference on Robotics and Biomimetics (ROBIO 2015)*, China, 2015, pp. 2359–2364.

[5] H. Moravec and A. Elfes, "High resolution maps from wide angle sonar," in *1985 IEEE International Conference on Robotics and Automation (ICRA 1985)*, USA, 1985, pp. 116–121.

[6] F. Ducho, A. Babinec, M. Kajan, P. Beo, M. Florek, T. Fico, and L. Juriica, "Path planning with modified a star algorithm for a mobile robot," *Procedia Engineering*, vol. 96, pp. 59–69, Dec. 2014.

[7] S. M. Lavalle, "Rapidly-exploring random trees: A new tool for path planning," Tech. Rep., 1998.

[8] A. Hornung, K. Wurm, M. Bennewitz, C. Stachniss, and W. Burgard, "Octomap: An efficient probabilistic 3D mapping framework based on octrees," *Autonomous Robots*, vol. 34, pp. 189–206, Apr. 2013.

[9] S. Niijima, Y. Sasaki, and H. Mizoguchi, "Autonomous navigation of electric wheelchairs in urban areas on the basis of self-generated 2D drivable maps," in *2018 IEEE/ASME International Conference on Advanced Intelligent Mechatronics (AIM 2018)*, New Zealand, Jul. 2018, pp. 1081–1086.

[10] Y. S. Shun Niijima and H. Mizoguchi, "Real-time autonomous navigation of an electric wheelchair in large-scale urban area with 3D map," *Advanced Robotics*, vol. 33, no. 19, pp. 1–13, Jul. 2019.

[11] K. Konolige, E. Marder-Eppstein, and B. Marthi, "Navigation in hybrid metric-topological maps," in *2011 IEEE International Conference on Robotics and Automation (ICRA 2011)*, China, 2011, pp. 3041–3047.

[12] F. Blochliger, M. Fehr, M. Dymczyk, T. Schneider, and R. Siegwart, "Topomap: Topological mapping and navigation based on visual SLAM maps," in *2018 IEEE International Conference on Robotics and Automation (ICRA 2018)*, Australia, 2018, pp. 3818–3825.

[13] M. Haklay and P. Weber, "OpenStreetMap: User-generated street maps," *IEEE Pervasive Computing*, vol. 7, no. 4, pp. 12–18, Oct. 2008.

[14] T. Ort, L. Paull, and D. Rus, "Autonomous vehicle navigation in rural environments without detailed prior maps," in *2018 IEEE International Conference on Robotics and Automation (ICRA 2018)*, Australia, 2018, pp. 2040–2047.

[15] M. Hentschel and B. Wagner, "Autonomous robot navigation based on openstreetmap geodata," in *13th International IEEE Conference on Intelligent Transportation Systems (ITSC 2010)*, Portugal, 2010, pp. 1645–1650.

[16] E. C. M. Pereira, D. A. Lima, and A. C. Victorino, "Autonomous vehicle global navigation approach associating sensor based control and digital maps," in *2014 IEEE International Conference on Robotics and Biomimetics (ROBIO 2014)*, Indonesia, 2014, pp. 2404–2409.

[17] P. Fleischmann, T. Pfister, M. Oswald, and K. Berns, "Using OpenStreetMap for autonomous mobile robot navigation," *Advances in Intelligent Systems and Computing*, vol. 531, pp. 883–895, Feb. 2017.

[18] S. Niijima, J. Nitta, Y. Sasaki, and H. Mizoguchi, "Generating 3D fundamental map by large-scale SLAM and graph-based optimization focused on road center line," in *IEEE International Symposium on Robot and Human Interactive Communication (RO-MAN 2017)*, Portugal, 2017, pp. 1188–1193.

[19] G. Floros, B. van der Zander, and B. Leibe, "OpenStreetSLAM: Global vehicle localization using openstreetmaps," in *2013 IEEE International Conference on Robotics and Automation (ICRA 2013)*, Karlsruhe, Germany, May 2013, pp. 1054–1059.

[20] M. A. Brubaker, A. Geiger, and R. Urtasun, "Lost! leveraging the crowd for probabilistic visual self-localization," in *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, USA, Jun. 2013, pp. 3057–3064.

[21] O. Vysotska and C. Stachniss, "Improving SLAM by exploiting building information from publicly available maps and localization priors," *Photogrammetry, Remote Sensing and Geoinformation Science*, vol. 85, no. 1, pp. 53–65, Feb. 2017.

[22] G. Grisetti, C. Stachniss, and W. Burgard, "Improved techniques for grid mapping with rao-blackwellized particle filters," *IEEE Transactions on Robotics*, vol. 23, no. 1, pp. 34–46, Feb. 2007.

[23] J. Zhang and S. Singh, "LOAM: Lidar odometry and mapping in real-time." in *Robotics Science and Systems*, USA, 2014, pp. 109–111.

[24] D. B. Johnson, "A note on Dijkstra's shortest path algorithm," *ACM*, vol. 20, no. 3, pp. 385–388, Jul. 1973.

[25] W. Hess, D. Kohler, H. Rapp, and D. Andor, "Real-time loop closure in 2D LIDAR SLAM," in *2016 IEEE International Conference on Robotics and Automation (ICRA 2016)*, Sweden, May 2016, pp. 1271–1278.