Proceedings of the SICE Annual Conference 2017
September 19-22, 2017, Kanazawa University, Kanazawa, Japan

A Hybrid Topological Mapping and Navigation Method for Large Area Robot Mapping

Ankit A. Ravankar[1†], Abhijeet Ravankar[1], Takanori Emaru[1], Yukinori Kobayashi[1]

[1]Faculty of Engineering, Laboratory of Robotics & Dynamics, Hokkaido University, Sapporo, Japan
(E-mail: ankit@eng.hokudai.ac.jp)

Abstract: In this paper, we present a hybrid topological mapping and navigation method for mobile robots. The proposed method combines metric and topological information to create map and generate navigation plan for the robot. As compared to traditional approaches of robot mapping, the method is lightweight and can be used for mapping and navigation in large areas which is particularly useful for service robots operating in large buildings. The method only uses local information for navigation while maintaining the global topological graph nodes. The topological nodes are used effectively for navigation and can also be used to store semantic information of the scene such as robot poses, scans and scene properties for complete long term robot autonomy. By combining the information from the two maps (topological and grid map), autonomous navigation and mapping in large areas for robots is possible.

Keywords: Robot Mapping; Topological Mapping; Navigation; Graph Theory; SLAM; Mobile Robot.

1. INTRODUCTION

The SLAM or simultaneous localization and mapping is an important problem to achieve autonomous behaviour for a mobile robot. A robot moving in an unknown environment needs to map the environment as it explores it. The robot perceives the information from the environment such as lines, obstacles, objects etc [1] using sensors attached to it such as lidars, camera or ultrasonic sensors. These information needs to be integrated into the map to make sense of the entitities. This problem is difficult as the robot needs to perform two tasks at the same time. Firstly, to localize itself in the given map and secondly, to construct the map of the environment while localizing. Thus there are two problems that need to be solved simultaneously. The SLAM problem has been extensively studied by researchers in the past decade and many solutions to improve the SLAM has been proposed in the past [2–4]. The most common challenges with any SLAM problem is to control the errors accumulating during the process [5]. Generally, wheel odometry is utilized to estimate the robot position and movement. However, with time the odometry is not consistent with the global map due to errors from wheel slippage and type of floor the robot is operating on. This results in maps built by the robot to become metrically inconsistent with the real world or ground truth. Another problem is that of the loop closure problem. When mapping a large environment as the robot returns back to a previously visited area, the robot cannot find correspondence between the features it has previously seen. The large inconsistencies accumulate errors over the time and eventually results in loop closure failure. An incorrect loop closing detection jeopordises the consistency of the map. Moreover, in robot configurations with only range sensors, identifying loop closure becomes a hard problem. A possible solution to solve this inconsistencies would be to use GPS (Global Position Sensor) to give the ground truth position of the robot. Unfortunately, GPS cannot be utilized indoors and error accumulation can only be reduced a little. In indoor environments metric maps are generally popular as they give rich information of the surrounding (Eg. Grid maps [6] and line segment maps [1]). Such metric maps are produced using odometry and laser scans. Recently 3D sensors such RGBD sensors have been extensively used to create point cloud maps (Eg. Octomaps) [7]. These maps typically require large memory for storage and demands high level of computing. On the other hand, topological maps can store relevant information in points called as nodes. A common e.g. of topological map could be a subway map of a city where each node represents a subway station. Topological maps represent robot environments as graphs. Nodes in such graph correspond to distinct situations, places, or landmarks. The advantage of using topological maps for robot exploration are that the nodes need not be defined in world frame and require less memory for storage unlike grid maps where every pixel in the map needs to be stored. The nodes are typically connected by edges that define some relation between the two connected nodes. It can be said that edges can define direct navigable paths between two nodes. For robot mapping the topological nodes can even store information (poses, scans) to help in robot localization. Topological maps contain far lesser information than metric maps (grid maps) and are suitable for smaller maps. To overcome this issue, in this paper we present a hybrid topological mapping and navigation scheme that combines metric and topological information. This can be used to overcome some of the previously mentioned disadvantages.

In this paper, we present the method to produce large scale indoor maps using a hybrid method that combines metric and topological maps. The edges of the topological nodes are connected using pose graphs. The edges are odometric based and define constraint between the nodes. The poses are periodically optimized to avoid loop closure failure [8] [9]. Earlier works on topological SLAM

† Ankit A. Ravankar is the presenter of this paper.

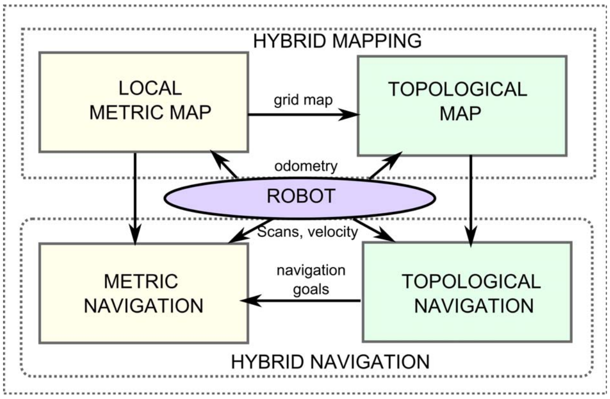

Fig. 1: Flowchart showing the proposed hybrid topological mapping method.

have focussed on involving semantic information in the map that includes object detection using RGBD sensors. Works by Pronobis et. al in [10] and [11] have extensively worked on the problem of fusing topological features into SLAM. However, most of the previous works have focussed on robots operating in small areas. Other works include the works presented in [12] and [13] that presents topological navigation for mobile service robots.

In this paper we also present, methods to perform navigation between topological nodes using standard path planning algorithms such as Djikstra and A-star algorithm. The advantage of using navigation on the topological nodes is that the nodes are already defined and can be used to generate the global path for navigation thus reducing the time required for calculating the global path. The edges and vertices can be used for direct navigation between subsequent or different nodes.

2. FRAMEWORK OF THE HYBRID TOPOLOGICAL MAPPING

Figure 1 presents the flowchart of the proposed hybrid topological mapping and navigation system. It consists of two main parts, the mapping system and the navigation system. Together they complete the hybrid system. The mapping system does localization and mapping and the navigation system does topological navigation along with metric navigation.

The local metric system provides grid map of the environment to the topological system which in turn runs its own topological SLAM. The grid map can be locally stored or instead be made as a global one. This depends on the requirement and application of the task at hand. For e.g. Just as humans perceive the environment, we tend not to remember each information of the places when we navigate. Humans only store fraction of information that is required to help them navigate even in complex environments. Information such as landmarks, intersections, shops etc helps human navigate efficiently. Similarly for the robot, only local information at times is necessary. When robot is operating in a small area, global information does not matter. All that is required is how the local map is made up and that helps it navigate

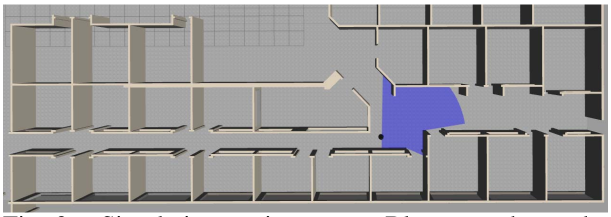

Fig. 2: Simulation environment. Blue area shows the lidar scans with robot at center.

while avoiding local obstacles. The maps are all centered around the robot, meaning the robot is considered to be the origin of the environment. In a way the robot can be asked to only store the local grid map with the robot at center, or to make a global map with all information. Storing only the local maps helps in saving considerable amount of memory. On top of this map the topological nodes are stored (globally). The nodes can be placed every meter of the robot motion or after several meters. For short distances and maps the inconsistencies does not exist and smaller number of nodes are utilized. However, as the size of the map increases its better to maintain nodes at smaller intervals in order to avoid any metrical inconsistencies and loop closure failures.

From the Fig. 1, the robot supplies relevant information such as the laser scans, odometry data to the local mapping system. The local mapping system generates a grid map and the topological systems builds the topological map on top of the grid map. When navigating in the environment whenever a goal location is received, the topological navigation system uses the topological map to find the optimum path. The optimum topological path is then passed to the metric navigation system in terms of small metric velocity goal commands. The local map and grid maps are used to avoid any collision with the obstacles (dynamic and static). In this way the whole system works in tandem with each other and the robot can be navigated easily to any point in the map.

Global Path Planning

The second sub-feature of the hybrid topological system consists of the navigation planner. There are different metric navigation planners such as the D-star path planner, A-star path planner, Djikstra and others. We utilized the A-star navigation that is found to perform good with grid maps. From the topological navigation system the shortest path around the graph is found using the A-star algorithm and the path is made. These path which passes through the nodes is then passed to the metric navigation system to perform navigation at the metric level. For path smoothing we adopted SHP path smoothing method as described in [14].

3. RESULTS

Graph Solving

Figure 4 shows how the graph is solved locally. Figure 4(a) shows the topological graph being constructed as

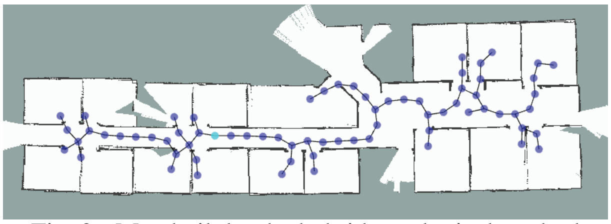

Fig. 3: Map built by the hybrid topological method

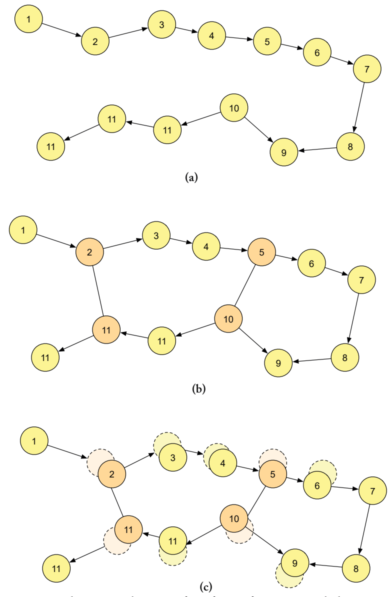

Fig. 5: Local graph optimization and loop closure

the robot explores from node 1 to node 11. Figure 4(b) shows the constraint edges between nodes 5 and 10 and node 2 and 11 after association has been made. The example shows the topological loop closure Fig 4(c) after the robot has traveled a certain distance. The solution to the graph constraints is performed for all nodes upto a fixed threshold distance from the associated node. The solution can be regarded as a least square problem which can be solved using the pose graph SLAM problem [8]. The solution to the problem can be found relatively faster as the nodes are limited as compared to full scale pose-graph problem. By solving the problem locally the robot can localize quickly and is always surrounded by nodes that can help it navigate reliably in the local map.

We show the results from our proposed method. Figure 2 shows the simulated environment with the robot. The environment was built using the open source Gazebo simulator. The robot used for the experiments is the Turtlebot with a fixed horizontal Hokuyo lidar. The Hokuyo lidar has a range of 4 m indoors and angular resolution of 240deg. The lidar scans are shown in blue color. The whole system was programmed and tested on ROS (Robot Operating System) on a Linux Ubuntu 14.04 x64 core i5 system. ROS provides basic libraries to perform mapping, localization and path planning. Figure 3 shows the final result of the mapping. Here, the topological nodes are represented by the blue circles. Nodes are placed every meter of the robot motion for better consistency between corresponding nodes and are connected using the odometric constraint. These are passed on top of the grid map that is generated from the metric mapper. The graph nodes are connected by edges and vertices that are optimized after every 1 sec using optimization algorithms (Levenberg Marquardt algorithm). Once the map and topological nodes are stored on the map, navigation can be performed by giving goal positions. Figure 5 shows the navigation planner when the robot is given a goal position to the top right of the room in the grid map. A plan is generated given by the green line. The planner tries to maneuver through the nodes while avoiding local obstacles. The robot follows the nodes to the goal point using the A-star algorithm in the proposed navigation planner.

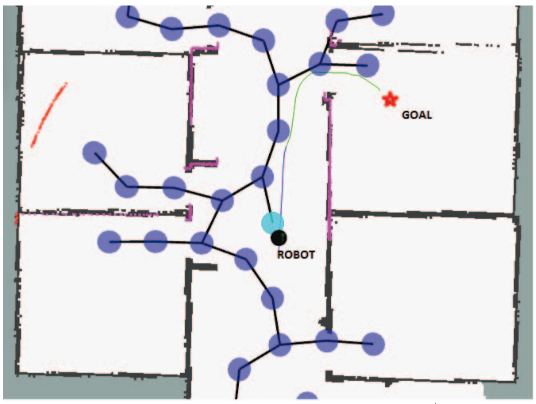

Fig. 4: Path planning on the nodes using A^* algorithm. Goal position is marked by red star. Green curve shows planned path by robot (shown as black circle).

The proposed method requires less memory, and is very lightweight algorithm to support autonomous robot mapping and localization on large scale. The proposed system is also very robust to errors occurring during the metric mapping. Table 1 shows the size of maps generated by the proposed system vs that by conventional GMapping [15, 16] package from ROS. From the table it is evident that the proposed hybrid approach requires far less memory to store the map in the form of topological nodes. The cumulative CPU time taken by the proposed algorithm as compared to the GMapping for large area mapping is shown in Figure 6. The proposed method is clearly more optimized for large scale mapping thus suitable for large scale mapping.

4. CONCLUSION

In this paper, we presented a novel hybrid topological mapping and navigation method for robot mapping. The system was successfully tested in simulation environment and the results were shown. The proposed system is lightweight and can be used for large scale maps for robot exploration. Moreover, the proposed algorithm is efficient and can run on smaller embedded processing boards easily. As the system benefits from both metric

Table 1: Comparison of size of maps generated using the proposed method

Size(KB)	Proposed	Gmapping [15]	Difference (%)
Map 1 (small size environment)	32	805	-96.02
Map 2 (large size environment)	84	3278	-97.43

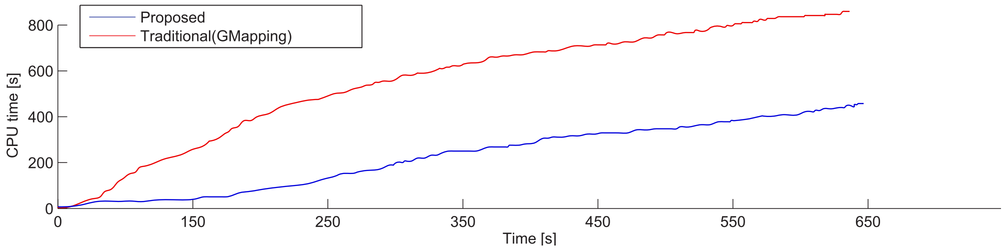

Fig. 6: CPU time - Proposed vs GMapping [15, 16]

and topological features, the proposed method is robust to indoor or outdoor environments. For future work, we plan to test the method on actual robot system. The current system lacks dexterity to correct loop closures as the environment for the simulations have very similar looking features (straight walls and rooms). We believe that testing in feature rich environment can solve this problem. In future, we plan to implement this system on a service robot for long term autonomy and extend the framework on multi-robot system for surveillance and monitoring applications. As a future work, we plan to utilize the semantic features from the environment using vision sensors and incorporate that into our mapping framework for complete autonomy.

REFERENCES

[1] A. Ravankar, A. A. Ravankar, Y. Hoshino, T. Emaru, and Y. Kobayashi, "On a hopping-points svd and hough transform-based line detection algorithm for robot localization and mapping," *International Journal of Advanced Robotic Systems*, vol. 13, no. 3, p. 98, 2016.

[2] M. Dissanayake, P. Newman, S. Clark, H. F. Durrant-Whyte, and M. Csorba, "A solution to the simultaneous localization and map building (slam) problem," *Robotics and Automation, IEEE Transactions on*, vol. 17, no. 3, pp. 229–241, 2001.

[3] M. Montemerlo and S. Thrun, "Simultaneous localization and mapping with unknown data association using fastslam," in *ICRA*. IEEE, 2003, pp. 1985–1991.

[4] A. A. Ravankar, Y. Hoshino, A. Ravankar, L. Jixin, T. Emaru, and Y. Kobayashi, "Algorithms and a framework for indoor robot mapping in a noisy environment using clustering in spatial and hough domains," *International Journal of Advanced Robotic Systems*, vol. 12, no. 3, p. 27, 2015.

[5] S. Thrun, W. Burgard, and D. Fox, *Probabilistic Robotics (Intelligent Robotics and Autonomous Agents)*. The MIT Press, 2005.

[6] S. Thrun, "Learning occupancy grid maps with forward sensor models," *Autonomous robots*, vol. 15, no. 2, pp. 111–127, 2003.

[7] C. Kerl, J. Sturm, and D. Cremers, "Dense visual slam for rgb-d cameras," in *Intelligent Robots and Systems (IROS), 2013 IEEE/RSJ International Conference on*. IEEE, 2013, pp. 2100–2106.

[8] N. Sünderhauf and P. Protzel, "Towards a robust back-end for pose graph slam," in *Robotics and Automation (ICRA), 2012 IEEE International Conference on*. IEEE, 2012, pp. 1254–1261.

[9] G. Grisetti, R. Kuemmerle, C. Stachniss, and W. Burgard, "A tutorial on graph-based SLAM," *Intelligent Transportation Systems Magazine, IEEE*, vol. 2, no. 4, pp. 31–43, 2010.

[10] A. Pronobis, O. M. Mozos, B. Caputo, and P. Jensfelt, "Multi-modal semantic place classification," *The International Journal of Robotics Research (IJRR), Special Issue on Robotic Vision*, vol. 29, no. 2-3, pp. 298–320, Feb. 2010.

[11] A. Pronobis and P. Jensfelt, "Large-scale semantic mapping and reasoning with heterogeneous modalities," in *Proceedings of the 2012 IEEE International Conference on Robotics and Automation (ICRA'12)*, Saint Paul, MN, USA, May 2012.

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

[13] H. Choset and K. Nagatani, "Topological simultaneous localization and mapping (slam): Toward exact localization without explicit localization," *IEEE Transactions on Robotics and Automation*, vol. 17, pp. 125–137, 2001.

[14] A. Ravankar, A. A. Ravankar, Y. Kobayashi, and T. Emaru, "Shp: smooth hypocycloidal paths with collision-free and decoupled multi-robot path planning," *International Journal of Advanced Robotic Systems*, vol. 13, no. 3, p. 133, 2016.

[15] R. O. S. Wikipedia. Ros- gmapping package. [Online]. Available: http://wiki.ros.org/gmapping

[16] 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, 2007.