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

Can Robots Help Each Other To Plan Optimal Paths in Dynamic Maps?

Abhijeet Ravankar†, Ankit A. Ravankar, Yukinori Kobayashi and Takanori Emaru

Faculty of Engineering, Lab of Robotics and Dynamics, Hokkaido University, Japan
(E-mail: abhijeet@frontier.hokudai.ac.jp)

Abstract: Mobile robots need to plan an optimal and collision-free path in complex environments to the various service locations in the map. Robot perception is limited to sensor range and hence robots are only aware of the obstacles within the range of its sensor. Moreover, the knowledge of new obstacles observed by each robot is kept local to itself. Due to these limitations, robots are often unaware of the new obstacles or path blockages in remote sections of the map. We proposed a knowledge sharing architecture in which the robots share the knowledge of new obstacles and blocked paths with each other. This enables the robots to update their map with the new and remote obstacles, and plan efficient paths with the timely information. This eliminates the need for re-planning of paths in traditional navigation methods. Experimental results show that the proposed method is efficient for multi-robot navigation in large and dynamic maps.

Keywords: Robot Path Planning, Shared Autonomy in Multi-Robot System, Multi-Robot Navigation System

1. INTRODUCTION

Mobile service robots are increasingly being used to automate many services at industries, hospitals, homes, and public places. Generally, multiple robots are employed as there are several advantages of task parallelism, diversity of tasks, and fault tolerance. The service areas of the mobile robots often vary with time. Hence, the robots need to plan optimal (for ex. shortest path) and collision free path from the current location (start) to their service location (goal).

In order to do path planning and navigation, a robot has to estimate its current location in the map. Moreover, a map is required in which the goal location is specified along with the traversable areas and obstacles. Mobile robots are equipped with a SLAM (Simultaneous Localization and Mapping) [1][2][3] module in order to perform localization (i.e. estimate its current location in the map) and mapping (building a map of the environment and update it).

Each robot keeps a pre-constructed map of the environment, and uses it for path planning. If new obstacles are found in the map, each robot updates its own map. The sensors attached to the robots have a limited range, and hence robot perception is limited only to the local areas in the sensor range. In a dynamic environment, there could be new obstacles in some remote areas of the map, or some obstacles might have changed their positions, whereas, some paths might have been blocked. This is not an uncommon scenario as, even at public places, some paths are purposefully blocked for cleaning purpose many a times. Paths can also be blocked due to repair work, or accidents. In traditional navigation methods, a robot can only know about the new obstacles or blocked paths of the environment by explicitly navigating those areas while observing the obstacles, and updating its own map. Robots are mostly unaware of the changes in the remote areas of the map which are outside their sensor's range. This *'locality of knowledge'*, in which, new obstacle information is only kept local to each robot, is a serious limitation in multi-robot systems, as each robot plans its path only by considering its own local map. Therefore, once the robot has navigated to the blocked location, it has to re-plan a new path to its goal location. In a large map with dynamic obstacles, path re-planning has to be done several times. This wastes a considerable amount of planning time and navigation.

We propose a knowledge sharing architecture, in which, if one robot finds a blocked path or new obstacles, it not only updates it own map, but also notifies other robots about the new information. This allows other robots to have a timely information about the remote obstacles, and they can plan their paths efficiently by considering the updated remote obstacle information. In the proposed scheme, each robot maintains its local map, and it can be updated with the obstacle information from other robots. This is particularly beneficial in large maps with many dynamic obstacles.

1.1. State of the Art

Many successful algorithms have been proposed for robot path planning. Among these, $A*$ algorithm [4], D* algorithm [5][6], probabilistic roadmap planner (PRM) [7], rapidly exploring random tree (RRT) [8][9], and potential fields [10] algorithms are widely used. Works in [11] and [12] provide a detailed summary of these algorithms. Multi-robot planning is either centralized or decentralized. In centralized approaches, all the paths of all robots are calculated simultaneously[13]. However, in decentralized approaches each robot calculates its path individually[14][15]. Work in [16] provide a scheme for multi-robot navigation in warehouses. Multi-Robot collision avoidance has been discussed in [17][18]. A review of multi-robot navigation strategies can be found in [20] [21]. While most of the previously proposed techniques in multi-robot path planning focus on strategies to avoid collision or collaboratively find an optimal path [18], the proposed work focuses on robots sharing information about the dynamic changes in the

† Abhijeet Ravankar is the presenter of this paper.

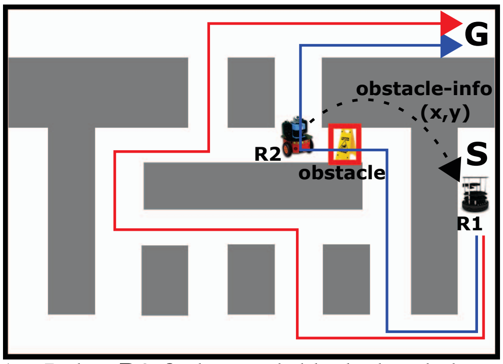

Fig. 1: Robot $\mathrm{R2}$ finds a path blocked and shares this information with $\mathrm{R1}$ enabling it to plan an optimal path shown in red color. Traditional method plan a path shown in blue color which requires re-planning.

environment to consider while planning their paths. Hence, any of the previously proposed path planning algorithm can be used with the proposed architecture.

2. NEW OBSTACLE UPDATE AND INFORMATION SHARING

The idea of sharing new obstacle information is graphically explained in Fig.1. Robot $\mathrm{R1}$ has a start location S, and goal location G. There is a new obstacle on the path which is observed by another robot $\mathrm{R2}$. In the traditional method, robot $\mathrm{R2}$ would update its local map with the information of the new obstacle, and this knowledge would be local to itself, leaving robot $\mathrm{R1}$ unaware of the new obstacle. In the traditional path planning, robot $\mathrm{R1}$ would plan the shortest route shown in blue color from S to G without considering the new obstacle as it is not in its sensor range. Robot $\mathrm{R1}$ would only discover the new obstacle by actually navigating to the obstacle location, and then re-plan its path towards the goal. On the other hand, in the proposed scheme, robot $\mathrm{R2}$ would not only update its own map, but would also broadcast the new obstacle information to other robots. Robot $\mathrm{R1}$ would update its map and plan an optimal path towards its goal shown in red color which avoids the new obstacle.

The state $[x \;\; y \;\; \theta]^T$ of the robot can be estimated using any of the probabilistic filters like Extended Kalman Filter (EKF), Unscented Kalman Filter (UKF), or Particle Filter (PF) [22], along with the uncertainty (Σ) associated with the pose. It is important to estimate the uncertainty as sensors are prone to errors which must be modeled. In the context of new obstacle update in the map, it is important because the uncertainty in obstacle location also needs to be modeled. Hence, if a new (say i^{th}) obstacle is updated, along with the absolute obstacle location (x^i_{obs}, y^i_{obs}), the uncertainty associated with it is also recorded (Σ^i_{obs}). Each robot is assigned a unique id (R_{id}). Moreover, is is assumed that all the robots are on the same network.

A grid based map of the environment is assumed to be available with each robot. For simplicity, it is also assumed that the same map is available with each robot so that there are no issues of map's scale and rotation differences. Using distance sensors like a laser range sensor, a robot can easily determine the approximate dimensions of the obstacle i.e. width (w^i) and breadth (b^i). For distance based sensors, it may be difficult to estimate the correct breadth, but only the width information is sufficient to determine if a path is blocked or traversable.

When a robot finds a new obstacle, it updates its map and broadcasts a message $\mathrm{M} = \{1, \mathrm{x}, \mathrm{y}, \Sigma, \mathrm{w}, \mathrm{b}, \mathrm{t}\}$ information to other robots. Here, t is the time at which the obstacle was observed. The prefix 1 in the message indicates that this obstacle is to be added to the map. Upon receiving this message, other robots update their maps with the new obstacle. In case of the grid maps, it is done by setting the grid pixels at location x, y with dimensions (w, b) to non-traversable values (zero in case of binary maps). The uncertainty information is incorporated by inflating the obstacle centered at (x, y) with Σ. Since Σ is a matrix, an eigenvalue-eigenvector decomposition of Σ is carried out which gives the direction and scale of variance. Similarly, if a robot finds that an obstacle has been removed, it broadcasts a message $\mathrm{M} = \{0, \mathrm{x}, \mathrm{y}, \Sigma, \mathrm{w}, \mathrm{b}, \mathrm{t}\}$. Here, the prefix 0 indicates that the obstacle at location (x, y) has been removed. The process of removing an obstacle in the grid map is similar as explained earlier.

With this update, other robots can plan an optimal path using any of the path planning algorithms described in Section 1.1. If the robot has already planned a path and started navigation, a check is performed if the new obstacle is in the way. If so, then the navigation is stopped and a new path is planned. In case of receiving an obstacle removal message, a check if performed if a shorter path is available from the current location. Thus, robots communicate and inform each other about the new obstacles in the map, enabling efficient path planning.

3. RESULTS

The proposed technique was tested in the simulation environment shown in Fig.3 using the Matlab software. D* algorithm [5][6] was used for path planning, however, any other path planning algorithm can also be used. A grid based navigation is chosen with one unit cost for forward, back, left, and right movement, whereas, for diagonal movement the cost is $\sqrt{2}$ units. In Fig.2, S and G represents the start and goal locations of the robot, respectively.

Figure 2(a) shows the path of Robot $\mathrm{R1}$ from location S to goal G, and it is the shortest path found by D* path planning algorithm. Figure 2(b) shows a new obstacle A found by a robot $\mathrm{R2}$ and its information is shared with $\mathrm{R1}$. Thus, $\mathrm{R1}$ plans an optimal path considering the new obstacle A. It is the optimal path given the knowledge of obstacle A. Figure 2(c) shows a scenario with two obstacles A and B found by robots $\mathrm{R2}$ and $\mathrm{R3}$. Robot

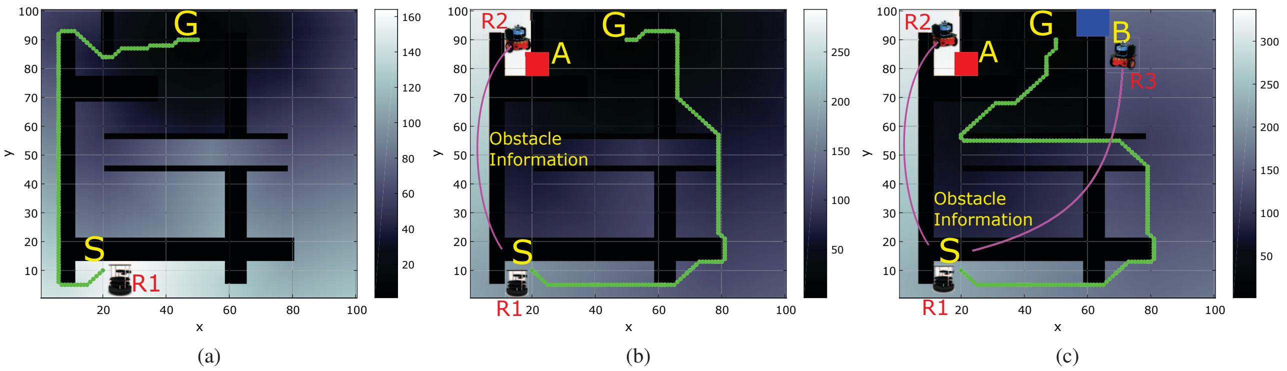

Fig. 2: D* planned paths from S to G. (a) Path without obstacles. (b) Path when knowledge of obstacle A is informed to robot. (c) Path when obstacle A and B are informed. The darker shades in the colormap represents proximity to goal (G).

Table 1: Path Planning Time in Traditional vs Proposed Method

Method	Start → Obstacle A	Obstacle A→ Obstacle B	Obstacle B→ Goal	(Total) Start→ Goal
Traditional	1.273 s	1.307 s	1.360 s	3.940 s
Proposed	–	–	–	1.317 s

Table 2: Total Navigation Distance in Traditional vs Proposed Method

Method	Start → Obstacle A	Obstacle A→ Obstacle B	Obstacle B→ Goal	(Total) Start→ Goal
Traditional	121.142	264.355	146.083	531.580
Proposed	–	–	–	211.296

R1 has timely information in this case about the remote obstacles and it plans an optimal path considering both the obstacles shown in Fig.2(c).

In traditional robot navigation without obstacle knowledge sharing, the robot R1 would have navigated the shortest path shown in Fig.2(a). Upon encountering obstacles A it would then re-plan another shortest path passing through obstacle B of which it has no information. It would then re-plan its path again near obstacle location B to the goal. Thus, it would require planning the path three times and results in more time spent in navigation and path planning which are avoided in the proposed scheme.

Table 1 and Table 2 shows the time required in path planning and the total distance navigated by the robot R1 in the traditional vs the proposed scheme in case of the scenario of two obstacles shown in Fig.2(c). The proposed method takes only 33% of the planning time and 39% of the distance is navigated by the robot as compared to traditional navigation. The saving in navigation distance also directly translates to saving battery power of the robots. Notice that, the first robot to find the obstacle needs to re-plan its path. However, subsequently, other robots benefit from the proposed shared autonomy.

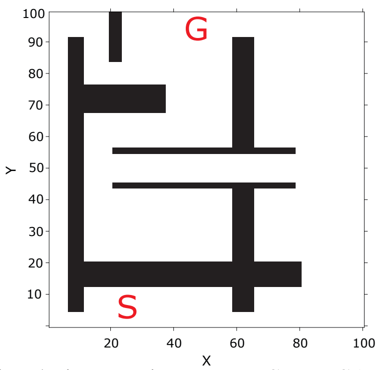

Fig. 3: Simulation Environment. Start: S(x=20,y=10), Goal: G(x=50,y=90). Black areas are obstacles.

4. CONCLUSION

This paper proposed an architecture in which robots can timely inform other robots about the new obstacles observed in the map. This allows the robots to update their maps and consider the new obstacles at remote locations while planning their maps. This eliminates, to a certain degree, the problem of local perception of the robots. The proposed method also eliminates the need to re-plan the path every time upon encountering a new obstacle which had already been found previously by another robot. The obstacle inclusion and removal was explained in case of grid maps. Simulation results show that the proposed knowledge sharing between robots is beneficial for efficient path planning compared to traditional robot navigation. The proposed scheme can bring significant performance gains in large and dynamic environments employing multiple service robots.

5. FUTURE WORKS

The proposed idea has many performance benefits in multi-robot navigation in large maps, nevertheless, there are many limitations which have not been addressed in the present work, and are considered as future works. First, is the problem of correspondence of location between two maps. Different robots may have maps of different scale and orientation. Thus, the problem of correspondence between maps needs to be addressed. The second problem is the problem of transience of obstacles. Generally, obstacles are not permanent in the map and therefore this transience needs to be modeled. Moreover, the proposed method needs to be tested in real environments too. We plan to address these important issues in future work.

REFERENCES

[1] 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, 2015.

[2] A. A. Ravankar, Y. Hoshino, T. Emaru, and Y. Kobayashi, "Map building from laser range sensor information using mixed data clustering and singular value decomposition in noisy environment," in *System Integration (SII), 2011 IEEE/SICE International Symposium on*, Dec 2011, pp. 1232–1238.

[3] 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.

[4] P. Hart, N. Nilsson, and B. Raphael, "A formal basis for the heuristic determination of minimum cost paths," *Systems Science and Cybernetics, IEEE Transactions on*, vol. 4, no. 2, pp. 100–107, July 1968.

[5] A. Stentz and I. C. Mellon, "Optimal and efficient path planning for unknown and dynamic environments," *International Journal of Robotics and Automation*, vol. 10, pp. 89–100, 1993.

[6] A. Stentz, "The focussed d* algorithm for real-time replanning," in *In Proceedings of the International Joint Conference on Artificial Intelligence*, 1995, pp. 1652–1659.

[7] L. Kavraki, P. Svestka, J.-C. Latombe, and M. Overmars, "Probabilistic roadmaps for path planning in high-dimensional configuration spaces," *Robotics and Automation, IEEE Transactions on*, vol. 12, no. 4, pp. 566–580, Aug 1996.

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

[9] S. M. LaValle and J. J. Kuffner, "Randomized kinodynamic planning," *The International Journal of Robotics Research*, vol. 20, no. 5, pp. 378–400, 2001. [Online]. Available: http://ijr.sagepub.com/content/20/5/378.abstract

[10] Y. Hwang and N. Ahuja, "A potential field approach to path planning," *Robotics and Automation, IEEE Transactions on*, vol. 8, no. 1, pp. 23–32, Feb 1992.

[11] S. M. LaValle, *Planning Algorithms*. Cambridge, U.K.: Cambridge University Press, 2006, available at http://planning.cs.uiuc.edu/ [Accessed: 11-02-2016].

[12] J.-C. Latombe, *Robot Motion Planning*. Norwell, MA, USA: Kluwer Academic Publishers, 1991.

[13] P. Svestka and M. H. Overmars, "Coordinated path planning for multiple robots," Department of Information and Computing Sciences, Utrecht University, Tech. Rep. UU-CS-1996-43, 1996.

[14] Y. Guo and L. Parker, "A distributed and optimal motion planning approach for multiple mobile robots," in *Robotics and Automation, 2002. Proceedings. ICRA '02. IEEE International Conference on*, vol. 3, 2002, pp. 2612–2619.

[15] 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.

[16] N. Pinkam, F. Bonnet, and N. Y. Chong, "Robot collaboration in warehouse," in *2016 16th International Conference on Control, Automation and Systems (ICCAS)*, Oct 2016, pp. 269–272.

[17] J. Stenzel and D. Luensch, "Concept of decentralized cooperative path conflict resolution for heterogeneous mobile robots," in *2016 IEEE International Conference on Automation Science and Engineering (CASE)*, Aug 2016, pp. 715–720.

[18] A. Ravankar, A. A. Ravankar, Y. Kobayashi, L. Jixin, T. Emaru, and Y. Hoshino, "An intelligent docking station manager for multiple mobile service robots," in *Control, Automation and Systems (ICCAS), 2015 15th International Conference on*, Oct 2015, pp. 72–78.

[19] T. Regev and V. Indelman, "Multi-robot decentralized belief space planning in unknown environments via efficient re-evaluation of impacted paths," in *2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, Oct 2016, pp. 5591–5598.

[20] A. Gasparetto, P. Boscariol, A. Lanzutti, and R. Vidoni, *Path Planning and Trajectory Planning Algorithms: A General Overview*. Cham: Springer International Publishing, 2015, pp. 3–27. [Online]. Available: http://dx.doi.org/10.1007/978-3-319-14705-5_1

[21] S. H. Tang, F. Kamil, W. Khaksar, N. Zulkifli, and S. A. Ahmad, "Robotic motion planning in unknown dynamic environments: Existing approaches and challenges," in *2015 IEEE International Symposium on Robotics and Intelligent Sensors (IRIS)*, Oct 2015, pp. 288–294.

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