Priority Exploration by Mobile Robots for Search and Rescue Situations

Arturo G. Roa-Borbolla, Antonio Marin-Hernandez, Ericka J. Rechy-Ramirez
Artificial Intelligence Research Center,
Universidad Veracruzana
Xalapa, Veracruz, Mexico, 91000
Email: roba200319 at gmail.com, [anmarin, erechy] at uv.mx

Hector Vazquez-Leal
Electronic Instrumentation and Atmospheric Sciences School,
Universidad Veracruzana
Xalapa, Veracruz, Mexico, 91000
Email: hvazquez at uv.mx

***Abstract*—Under disaster situations the time to rescue victims is as important as the lives of rescue teams. Autonomous mobile robots can be used to explore the dangerous areas in order to avoid exposing humans to these hazard situations. However, robots need to be intelligently programmed to explore efficiently the environment. Moreover, robots need to consider several conditions and knowledge of the environment to provide priority to more crowded spaces and ignore the unoccupied ones. This study introduces a priority path planning for search and rescue situations using mobile robots. A weighted (priority) graph representing a-priori information about the population of each space in a public building is used for intelligent path planning. Therefore, the resulting path is the shortest path with a maximum gain, i.e. fastest exploration and/or more information gain (victims detected), information that should be send to the rescue teams in order to proceed efficiently.**

I. Introduction

A search and rescue situation is a situation where an unexpected event happened and people's lives are compromised or in danger. Under these situations, search and rescue operations play a key role in order to provide aid to people whose need it without risking the lives of the rescue teams.

Nowadays, robots are used to accomplish some of these rescue tasks. Mobile robots have been used to explore places using SLAM (Simultaneous Localization and Mapping) techniques. The maps generated by the robots during the exploration of the places provide key information to rescuers; however, the generation of the maps might be time-consuming depending on the structure of the building. Moreover, in order to decide the next point or space to explore, common exploration techniques use only in-flight information obtained by the robot from the building already damaged.

There are some events where the integrity of the building has not been complete damaged, for instance: situations where there was a quake or a gas leak, and then the structure of the building has not been compromised. In order to explore these types of environments, robots need to focus their exploration tasks on crowded spaces and discarting the unoccupied ones. Consequently, robots can provide key information regarding the crowded spaces to the rescue teams, so that the search and rescue operations can be focused on these spaces.

Based on the fact that public buildings should have a map stored at the cloud of each floor or section indicating evacuation routes; robots can use this map in order to have a-priori information regarding the structure of the building to be explored. Therefore, robots might identified bathrooms, halls, administrative or general offices, so that they can identified crowded places and prioritize these places for rescue and search operations.

Exploration is the capability to choose where to go and navigate to a given point. A well exploration is defined by gathering as much useful information as possible with minimal movements and shorter time. Exploration could be used to create or navigate maps which reflects in some abstraction level the structure of the environment. In mobile robotics, there are several methods to explore a given area, e.g. Gap Navigation Trees, Potential Fields, Frontier Methods, Information Gain, etc.

Generally, maps are employed to represent the explored area. There are different types of maps, these can be classified as: metric, topological or hybrids. Recently hybrid maps are widely used in exploration because these maps merge the advantages presented in metric and topological maps (e.g. topological maps provide best precision about the positions of the robot and obstacles, as well as good path planing. Furthermore, hybrid maps are made using a topological map created over metric [1]–[3]).

This paper proposes a hybrid (metric - topological) map to solve the exploration problem in search and rescue situations with a-priori knowledge. Unlike previous studies presented in the literature, our approach uses priority information, which is added on nodes of topological map in order to get the maximum gain in the route. Moreover, this hybrid map is created from the combination of: i) a metric map, which it is used to indicate emergency exits, the concurrence, time utility and, ii) a topological map, which provides conditions for each individual space represented on graph. Based on this information, the robot is able to prioritize the search of victims without searching in every space, therefore, key information regarding the victims can be gathered promptly.

This paper is organized as follows. In section II some related studies regarding the state-of-the-art are explained. Section III introduces our approach. Section IV discusses the experiments and current results. Finally, section V presents our conclusions

978-1-5386-0819-7/17/$31.00

and future work.

II. Related Work

Potential fields have been used widely in path planning [4], [5]. This method has been employed for: planning paths over constructed maps and exploring environments while the map is being built. A metric map modeled as a discrete grid is the common environment representation [6], [7]. In this map, each of its cells with fixed size stores the occupancy probability, which can be discretized as 3 states: free, unknown and occupied.

First, a high value is assigned to the start point of the map, and a low value is given to the end point of the map (frontier zones). Then, a descendant gradient algorithm through the empty cells of the map is employed in order to navigate from one point to another using the shortest path (lowest cost) and avoiding obstacles.

Due to the fixed size of each cell, according to [8], the distance between two points can be known by multiplying the number of cell between both points by the cell size in every dimension.

Different methods to construct graphs over metric maps have been proposed. This type of maps can be created through SLAM, i.e. while sensors are collecting data, a metric map is constructed to represent the environment and a topological map is constructed over metric [9]. Another approach is that a topological map is constructed given a metric map already created [10]. In [11], authors propose an approach to construct a topological map over a metric one through the definition of areas of visibility represented as nodes. The navigation from one area to other is performed by connecting areas through doors.

Regarding topological methods, these methods performed the exploration using connectivity graphs. In this case, once a node has been reached, there is no need to measure distance or orientation. A topological map is a graph of the environment in which path planning can be done easily. Once a node has been reached, this is marked as visited place. Consequently, metric and orientation measurements are calculated only when the reached node is marked as no-visited place. In this type of maps, routes are known through the connections of two or more nodes. Moreover, these connections are invariant, thus there is no need to compute the path every time. The only data required to calculate the path is the current node and the adjacent ones. An approach defined in [12] can identify specific places in a map and distinguish between a visited place and no-visited place.

A topological map has a more complex approach where restrictions are considered, this type of maps are illustrated as weighted graphs. The edges of these graphs have associated costs (weight). Therefore, constraints might be implemented in this type of graph. An approach using constrains can be found in [13].

In this type of graph the lowest cost is computed instead of the shortest path, i.e. the lowest cost path could or could not be the shortest one and could have more nodes than the shortest path. The function of cost is implemented using the edges.

As mentioned earlier, there is a special interest regarding the metric-topological maps (hybrid maps) due to its precision and simplicity. Both characteristics are useful in robotics, specially in path planning because there is the need to navigate through spaces avoiding obstacles and getting maximum information and the lowest cost path.

In literature, strategies of path planning are based mostly on the cost to reach a node in order to find the shortest path. In this context, Dijkstra, A* or D* algorithms [14] are widely used on grid based maps and graphs to find the path with the lowest cost. These methods try to search the shortest path from a point A to point B. Additionally, in robotics, it is desirable that the method involves constrains on the path [13] (e.g. obstacle or risk avoidance).

Nevertheless, finding the shortest path is not the only problem that robots have to face, there are other constrains or scenarios to consider when the distance is not the only measure to take into account. Therefore, the previous algorithms are not suitable [13]. In this case, weighted graphs or constrains graphs are useful. Moreover, multi-dimension weighted [15] or multi-constrains [16] graphs have been proposed to plan the path and exploration.

Due to compactness and invariant structure of topological maps, planning using these maps is more effective than using grids. Therefore, it might be less time-consuming even if the shortest path is not the only measurement to consider. The computation of the shortest and the lowest-cost path depends only on the number of nodes (areas) defined in the graph and not on the world size which is much larger.

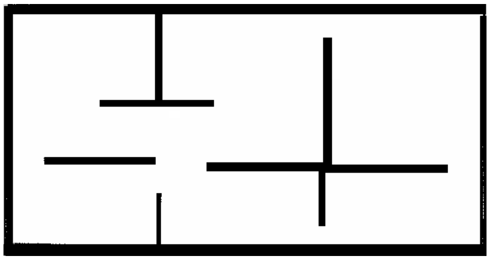
Path planning is used in search and rescue situations to explore the spaces. Specifically, SLAM technique is employed in unknown environments to perform the path planning because the structure of the building might present inaccuracies due to the damage [17]–[19]. Generally, it is desirable that tasks on this type of scenarios are autonomous, semi-autonomous or collaborative [20], and they must be directed to the most promising regions of the environment [19].

As discussed above, exploration methods have been widely used in the literature to perform SLAM. For instance, a visibility map has been created using a guard robot with a pre-build map [21].

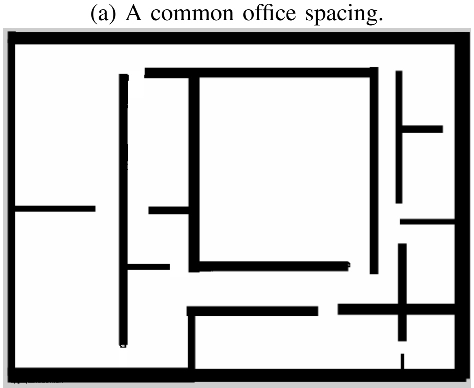
III. Priority Path Planning Exploration

As mentioned earlier, our approach uses hybrid maps. Table 1 shows two of the maps employed in the experiments. These maps were chosen because they are similar to scenarios of common offices.

Usually, the representation of the environment is performed via metric maps. Consequently, the proposed hybrid map is created based on the metric one using the method described in [11]. In this method, every node is defined by a visibility area. A slight difference is that on map constructed connection nodes are added in the middle of every shared area border (Doors - Connections), then every area node (An) is connected

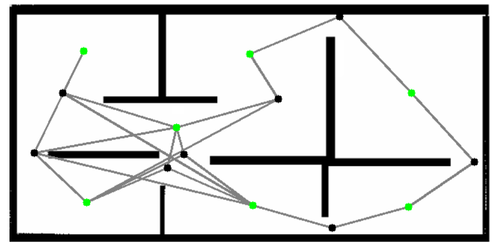


(a) A common office spacing.

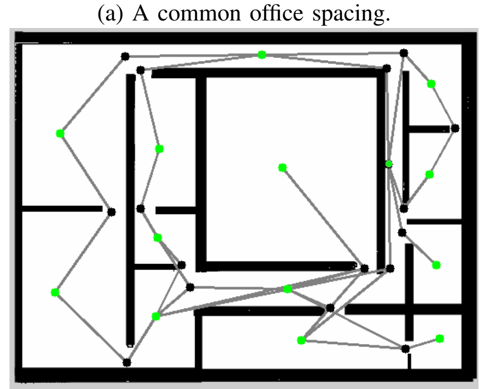


(b) A modification of our working environment.

Fig. 1: Metric Maps. a) A common office spacing b) A modification of our working environment.



(a) A common office spacing.



(b) A modification of our working environment.

Fig. 2: Hybrid Maps. Topological maps constructed over metric. a) A common office spacing and b) modification of our working environment. Green; Area Nodes. Black: Connection Nodes. Gray: Connection between Nodes.

with every connection node (Cn) if a line of sight exists (such as in table 2).

Then, every node has a position on the metric map and there are N nodes where:

$$N = An + Cn \tag{1}$$

Once connection has been done, a priority (gain) is assigned to every area node (connection nodes has no priority assigned), then a gain node list $G = [g_1, g_2, ..., g_i]$ is created and a gain g_i for every zone is calculated and assigned initially through:

$$g_i = \frac{100}{\sum_{i=1}^{An} i} * An - i \tag{2}$$

In order to calculate the cost, the metric map is used because every node has a position x, y in a 2D metric map and every connection is done through the line of sight. There is a line between nodes and its distance can be calculated through its points. Based on this, a path P that crosses all or mostly of area nodes without repeating from first to last area node in graph is calculated, then cell cost is computed through:

$$CL = \frac{100}{D} \tag{3}$$

where:

$$D = \sum_{i=1}^{N-1} \sqrt{(P_{i+1}(x) - P_i(x))^2 + (P_{i+1}(y) - P_i(y))^2} \tag{4}$$

This ensures that lost by cell becomes invariant and the cost between nodes can be calculated by computing the nodes that need to be crossed in a path and their distances. Once CL is calculated, every possible path of size Ps from current nearest node to maximum priority node that do not repeat area nodes is calculated. Then, cost of every path is computed by:

$$L = S * CL \tag{5}$$

where:

$$S = \sum_{i=1}^{Ps-1} \sqrt{(P_{i+1}(x) - P_i(x))^2 + (P_{i+1}(y) - P_i(y))^2} \tag{6}$$

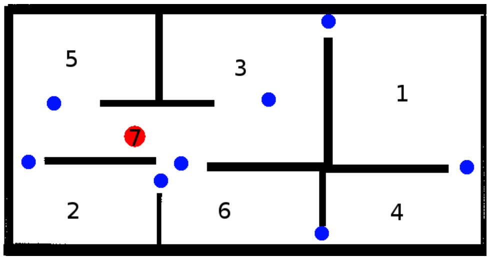
To calculate gain in a route is just needed to add every individual node gain g_i in path as:

$$Gr = \sum_{i=1}^{An} g_i \in P \tag{7}$$

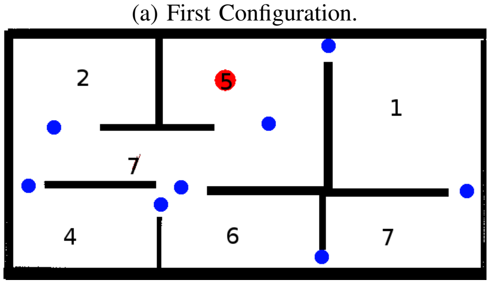
Next, every possible route from nearest node to maximum priority area without repeating area nodes is calculated and a list of paths $LP = [P_1, P_2, ..., P_i]$ is created and for every P_i its Gr_i is calculated and add to a list of gains Grl.

Then, the Best Route to Max priority node is calculated by 8:

$$Br = \arg\max\left(\frac{Grl}{L}\right) \tag{8}$$



(a) First Configuration.



(b) Second Configuration.

Fig. 3: a) First and b) second configuration of office spacing. Numbers represents position and priority of every area node, blue points are connection nodes and red points are the Starting Area Nodes.

Once the first node is reached a list V is initialized and, every time a node is reached An_i is added, then a priority increment is calculated by:

$$PI = \frac{\sum_{i=1}^{Vn} g_i}{Vn} \tag{9}$$

for each $g_i \in V$ and a priority update is done by $g_i = g_i + PI$ if $g_i \notin V$ and for each $g_i \in V$ $g_i = 0$

IV. Experiments

As mentioned above, various experiments were performed using two different maps. In order to do experiments on Gazebo (ROS simulator) with a simulated robot (a P3DX from Adept Mobile Robots), these two maps were created using ROS. These environments are modified slightly from our real building and some office spacing used by university researchers. The first one (table 3) is a simply common office and the second one is a modification of our current working building (Fig. 4).

As indicated previously, the first world represents a simply office world in where two configurations with different starting points were tested. In the first configuration the priority nodes were defined as (from left to right and top to bottom): 5, 3, 1, 7, 2, 6, 4. In this configuration, the node 7 was defined as initial node and since node 1 is the highest priority node it was selected as the finishing node.

In the second configuration the graph was defined as 2, 5, 1, 7, 4, 6, 3. In this configuration node 5 was selected as the initial node and as in test 1, node 1 is the finishing node. The results are presented in next sections 1) and 2).

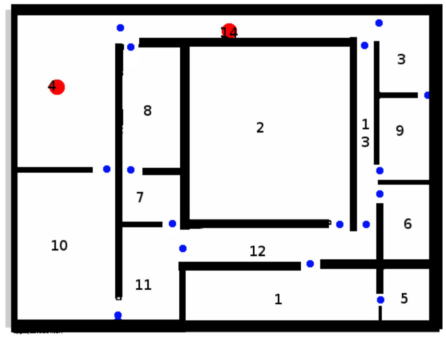


Fig. 4: Configuration of author's working space, numbers represents position and priority of every area node, blue points are connection nodes and red points are the Starting Area Nodes.

1) Test 1: In test 1, the cell loss obtained was $CL = 0.034$, and because the area node was 7, the gain obtained for every node was $G = [28.571, 23.809, 19.047, 14.285, 9.523, 4.761, 0.0]$. Then, if the shortest path is from $An = 7$ to $An = 1$, the resulting path is: $sP = [7, 12, 3, 8, 1]$. If connection node is eliminated and gain of every area node is added, the gain for this route is obtained by: $Gr_{sP} = 47.619$. Again, through metric map, loss in route is calculated: $L = 52.007$.

Then, dividing the gain in the route by the total lost, the shortest path gain is calculated as: $PGr = 0.915$. To compute Br through priority weighted graph, it is added every gain of area nodes in : $BrP = [7, 11, 2, 12, 3, 8, 1]$, then: $Gr_{BrP} = 71.428$.

The loss in the best route is calculated as: $L = 67.375$, and the gain in the best calculated route is: $Br = 1.060$.

As explained earlier, the algorithm of Br decides to go first to node 2 because the priority and closeness obtain a higher gain than in the shortest path. By choosing this path Br, algorithm chooses a path balanced between gain and distances adding 23.809 to Gr and 15.367 to L.

2) Test 2: As in test 1 the number of area node was 7, in this scenario the priority of nodes was the only one changing, the cell loss and the list of node gain remain as $CL = 0.034$, and the gain list is $G = [28.571, 23.809, 19.047, 14.285, 9.523, 4.761, 0.0]$. Then, if the shortest path from $An = 5$ to $An = 1$ is calculated, the resulting path is: $sP = [5, 8, 1]$. And the gain for the shortest path route is: $Gr_{sP} = 38.095$. The calculated loss for this path is: $L = 27.019$.

And dividing the route gain by the path loss, the path gain is obtained as: $PGr = 1.409$. To calculate Br through priority weighted graph, every gain of nodes is added in: $BrP = [5, 8, 1]$. Then gain for the best route is:$Gr_{BrP} = 38.095$. And the loss for the best route is: $L = 27.019$. And calculating the best route gain by: $Br = 1.409$. The best route and the shortest path match. This means that gain of near node is not high enough to deviate the path followed by the robot.

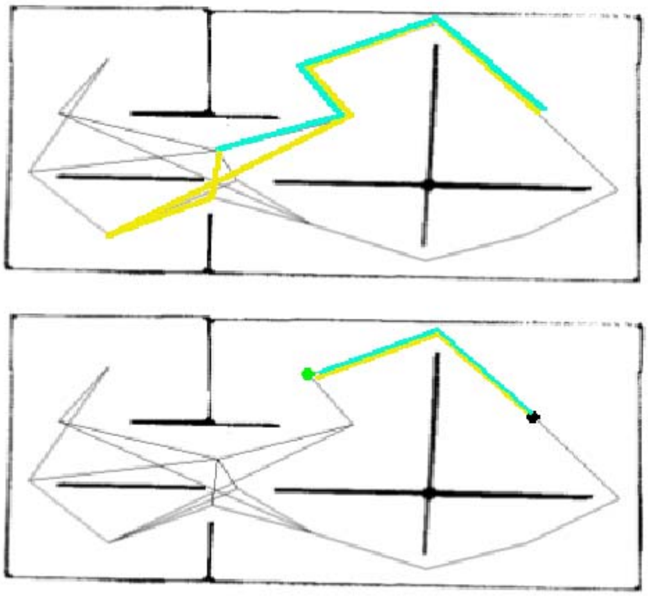


Fig. 5: Results of test 1 (Top) and 2 (Bottom) on common office spacing world. Green point: Initial position. Black point: Finish node. Yellow line: Best Route. Cyan line: Shortest Path.

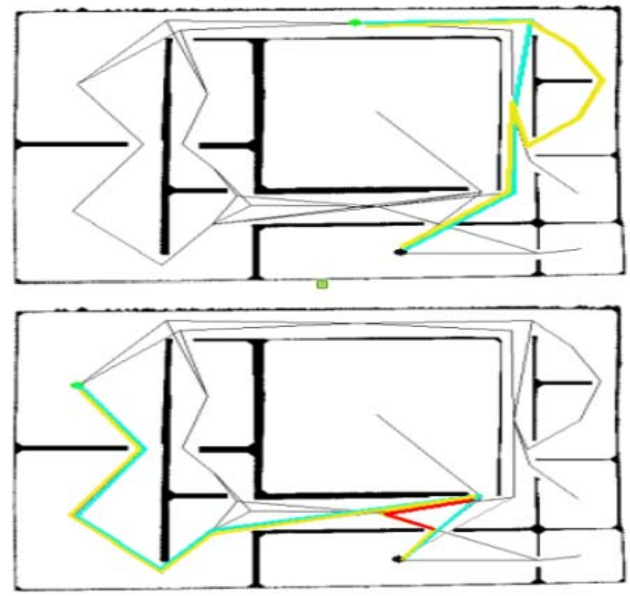


Fig. 6: Results of test 1 (Top) and 2 (Bottom) in modified working world. Green point: Initial position. Black point: Finish node. Yellow line: Best Route. Cyan line: Shortest Path. Red line: Path added

A. Modification of our working environment

World used in this experiment is a modification of author's working building. In this experiment, only one configuration was defined. Area nodes were defined as (from left to right and top to bottom): $4, 14, 3, 8, 2, 13, 9, 10, 7, 6, 11, 12, 1, 5$. Then, the first selected starting point was 14 and the second one was the node number 4. In both cases, the finish point was 1 due to its high priority. Results are shown in next sections 1) and 2).

1) Test 1: In the first test the cell lost obtained were $CL = 0.019$ and the gain list was $G = [14.285, 13.186, 12.087, 10.989, 9.890, 8.791, 7.692, 6.593, 5.494, 4.395, 3.296, 2.197, 1.098, 0.0]$. if the shortest path is from $An = 14$ to $An = 1$, the resulting path is: $sP = [14, 31, 13, 27, 1]$. Therefore, the gain in the route adding every area node in path is: $Gr_{sP} = 31.301$. And the lost for this path is: $L = 13.393$. Then, dividing the gain of nodes in route by the lost of the route, the shortest path is calculated as: $PGr = 2.337$.

To compute Br through priority weighted graph, every area node in $BrP = [14, 29, 3, 28, 9, 27, 13, 24, 1]$ is added, then the gain $Gr_{BrP} = 49.198$ is obtained and the lost for this route is calculated as: $L = 16.852$. Again dividing the node gain by the lost, the best-route gain is: $Br = 2.919$

It can be seen that a gain of 2.919 is obtained when the best route algorithm is used; whereas a gain of 2.337 is reported when the shortest path algorithm is applied. Consequently, it can be concluded that the best route algorithm found the best way to travel along the path with maximum gain. On the other hand, the shortest path algorithm found the fastest way to go from node 14 to 1, but in a long-term many repetitions of nodes are done and some priority nodes are dropped or delayed.

2) Test 2: Regarding the office scenario, the cell lost and list of gain remain in the second test as $CL = 0.019$ and G = [14.285, 13.186, 12.087, 10.989, 9.890, 8.791, 7.692, 6.593, 5.494, 4.395, 3.296, 2.197, 1.098, 0.0]. If the shortest path from $An = 4$ to $An = 1$ is calculated through the path $sP = [4, 15, 6, 16, 11, 22, 1]$, the gain of nodes for this route is $Gr_{sP} = 47.353$ and the lost of the path is $L = 15.490$. As in previous tests, the gain for the route is obtained dividing the gain by the lost: $PGr = 3.056$. To calculate Br through priority weighted graph in: $BrP = [4, 15, 6, 16, 11, 21, 12, 23, 1]$, the gain is obtained as: $Gr_{BrP} = 47.360$ and lost is calculated as: $L = 13.672$. Then the best route gain is obtained as: $Br = 3.463$

It can be seen that in both scenarios the best route algorithm added one or more nodes to the shortest path. In second test, the best route algorithm added just one area node defined in the corridor because of its gain and its closeness, adding more gain than its lost.

V. Conclusion

This paper presented a modified version of the path planning with visual graph, which considers constraints or priorities over a hybrid map. Priorities determine the population of certain spaces (i.e. if the space is crowded). This is useful when robots explore environments in order to avoid wasting time in the rescue mission. Consequently, the search might be focused only on the crowded spaces, by ignoring the empty spaces.

Robots can modify the path planning through the use of priorities assigned to crowded spaces, therefore, the path with the highest gain is employed. Finally it could be concluded that in both worlds (modification of our real working environment and simply office), novel approach of priority weighted graph path planning demonstrated very good performance. In future work, an implementation of this modified algorithm will be tested in real situations as well as compared with some other approaches despite of their differences.

REFERENCES

[1] A. Marin-Hernandez and M. A. Navarrete, *Topological-Metrical Map Creation and Path Planning for Autonomous Indoor Mobile Robots*. SMIA, 2008, pp. 71–83.

[2] S. Thrun, "Learning metric-topological maps for indoor mobile robot navigation," *Artificial Intelligence*, vol. 99, no. 1, pp. 21 – 71, 1998. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0004370297000787

[3] B. Kaleci, . M. Senler, O. Parlaktuna, and U. Grel, "Constructing topological map from metric map using spectral clustering," in *2015 IEEE 27th International Conference on Tools with Artificial Intelligence (ICTAI)*, Nov 2015, pp. 139–145.

[4] C. I. Connolly, "Applications of harmonic functions to robotics," in *Proceedings of the 1992 IEEE International Symposium on Intelligent Control*, Aug 1992, pp. 498–502.

[5] C. I. Connolly, J. B. Burns, and R. Weiss, "Path planning using laplace's equation," in *In Proceedings of the 1990 IEEE International Conference on Robotics and Automation*, 1990, pp. 2102–2106.

[6] A. Elfes, "Sonar-based real-world mapping and navigation," *IEEE Journal on Robotics and Automation*, vol. 3, no. 3, pp. 249–265, June 1987.

[7] W. K. Yeap and M. E. Jefferies, "Computing a representation of the local environment," *Artificial Intelligence*, vol. 107, no. 2, pp. 265 – 301, 1999. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0004370298001118

[8] E. P. e Silva, P. M. Engel, M. Trevisan, and M. A. Idiart, "Exploration method using harmonic functions," *Robotics and Autonomous Systems*, vol. 40, no. 1, pp. 25 – 42, 2002. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0921889002002099

[9] 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.

[10] A. G. Ozkil, Z. Fan, J. Xiao, J. K. Kristensen, S. Dawids, K. H. Christensen, and H. Aanaes, "Practical indoor mobile robot navigation using hybrid maps," in *2011 IEEE International Conference on Mechatronics*, April 2011, pp. 475–480.

[11] W. Yeap and M. Jefferies, "Computing a representation of the local environment," *Artificial Intelligence*, vol. 107, 1999.

[12] B. Kuipers and Y.-T. Byun, "A robot exploration and mapping strategy based on a semantic hierarchy of spatial representations," *Robotics and Autonomous Systems*, vol. 8, no. 1, pp. 47 – 63, 1991, special Issue Toward Learning Robots. [Online]. Available: http://www.sciencedirect.com/science/article/pii/092188909190014C

[13] Q. Sun, W. Wan, G. Chen, and X. Feng, "Path planning algorithm under specific constraints in weighted directed graph," in *2016 International Conference on Audio, Language and Image Processing (ICALIP)*, July 2016, pp. 635–640.

[14] S. Narayan, T. V. Prasannakumar, M. Prasanth, N. P. Velavan, N. P. Kumar, and A. S. Pillai, "A priority based exploration algorithm for path planning of an unmanned ground vehicle," in *2014 International Conference on Embedded Systems (ICES)*, July 2014, pp. 275–280.

[15] S. Jiang, Z. Feng, X. Zhang, X. Wang, and G. Rao, *A Multi-dimension Weighted Graph-Based Path Planning with Avoiding Hotspots*. Singapore: Springer Singapore, 2016, pp. 15–26.

[16] Z. Wang and T. Wang, "A heuristic algorithm for shortest path with multiple constraints," in *International Conference on Communication Technology Proceedings, 2003. ICCT 2003.*, vol. 1, April 2003, pp. 482–486 vol.1.

[17] D. Calisi, A. Farinelli, L. Iocchi, and D. Nardi, "Autonomous exploration for search and rescue robots," in *Proceedings of the International Conference on Safety and Security Engineering (SAFE)*, 2007, pp. 305–314.

[18] A. Kleiner, J. Prediger, and B. Nebel, "Rfid technology-based exploration and slam for search and rescue," in *2006 IEEE/RSJ International Conference on Intelligent Robots and Systems*, Oct 2006, pp. 4054–4059.

[19] A. Carbone, D. Ciacelli, A. Finzi, and F. Pirri, *Autonomous Attentive Exploration in Search and Rescue Scenarios*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2007, pp. 431–446.

[20] V. Govindarajan, S. Bhattacharya, and V. Kumar, *Human-Robot Collaborative Topological Exploration for Search and Rescue Applications*. Tokyo: Springer Japan, 2016, pp. 17–32.

[21] I. Ardiyanto and J. Miura, "Visibility-based viewpoint planning for guard robot using skeletonization and geodesic motion model," in *2013 IEEE International Conference on Robotics and Automation*, May 2013, pp. 660–666.