https://doi.org/10.2352/EI.2022.34.16.AVM-117

Sensor-aware Frontier Exploration and Mapping with Application to Thermal Mapping of Building Interiors

Zixian Zang, Haotian Shen, Lizhi Yang, Avideh Zakhor. Univeristy of California, Berkeley

Abstract

The combination of simultaneous localization and mapping (SLAM) and frontier exploration enables a robot to traverse and map an unknown area autonomously. Most prior autonomous SLAM solutions utilize information only from depth sensing devices. However, in situations where the main goal is to collect data from auxiliary sensors such as thermal camera, existing approaches require two passes: one pass to create a map of the environment and another to collect the auxiliary data, which is both time consuming and energy inefficient. We propose a sensor-aware frontier exploration algorithm that enables the robot to perform map construction and auxiliary data collection in one pass. Specifically, our method uses a real-time ray tracing technique to construct a map that encodes unvisited locations from the perspective of auxiliary sensors rather than depth sensors; this encourages the robot to fully explore those areas to complete the data collection task and map making in one pass. Our proposed exploration framework is deployed on a LoCoBot with the task to collect thermal images from building envelopes. We validate with experiments in a multi-room commercial building. Using a metric that evaluates the coverage of sensor data, our method significantly outperforms the baseline method with a naive SLAM algorithm. The code can be found at https://github.com/lzyang2000/herox.

I. Introduction

Autonomous mapping of unknown environments is a well-known problem in the field of robotics. The autonomous mapping process involves localization, mapping, and exploration. In practice autonomous exploration algorithms work in tandem with simultaneous localization and mapping (SLAM) algorithms to map three dimensional spaces efficiently. Frontier based exploration technique is a frequently used autonomous exploration strategy of 2D and 3D environments whereby unexplored edges in maps are deemed frontiers and the robot is compelled to explore those. Current SLAM and frontier exploration algorithms typically only utilize depth sensors. However, in applications where the main task is to collect spatially registered auxiliary data from other sensors such as thermal camera, the existing approaches to data collection typically require two passes: in the first pass the frontier exploration algorithm guides the robot to construct a map and in the second pass, the robot traverses the space with the known map to collect auxiliary sensor data. Though simple and intuitive, this method is time consuming and energy inefficient, as it needs to traverse the environment twice. Current frontier exploration algorithms do not cater to the needs of data collection from auxiliary sensors in the SLAM process. We aim to develop a novel frontier exploration algorithm with task specific constraints, to simultaneously build a map of the environment and collect auxiliary data all in one pass.

Frontier based exploration algorithm, first proposed in [1], is a well known algorithm used in autonomous SLAM for path planning deciding which region to explore. As described in [1], a frontier is defined as the boundary between known and unknown areas of the map. It motivates the robot to select target locations that yield the most information gain (i.e. frontiers), thus improving the exploration efficiency. A frontier exploration algorithm mainly consists of two parts - choose a frontier goal point, then plan and execute a path to that goal point. The frontier goal point selection can be further broken down into 3 parts - locating the frontiers, generating candidate frontier goal points, and ranking those points for selection. The frontier localization process is akin to edge detection in computer vision, and can be done with edge detection algorithms. In [1], the authors utilize BFS in order to detect the frontiers more efficiently, as transforming the map to images and back is less efficient and not precise. After the frontier edges have been found, candidate points must be generated on these edges. Methods of generating these frontier points include first randomly sampling points with given offsets in both location and orientation, removing the points that fall in the unknown region and then using K-means clustering to group nearby points together as in [2]. Points can also be generated by choosing the centroid of the frontiers on resource-constrained systems. Then the points are ranked via a metric defined as the sum of uncertainty scores of the grid points between the robot and the frontier. Finally, A-star [3] and trajectory sampling-based [4] planners are used primarily for guiding the robot to the chosen frontier goal point.

II. Method

The need to autonomously explore a space while collecting auxiliary sensor data requires our proposed system to have four major components: SLAM, ray-tracing, exploration and actuation. The SLAM module uses depth sensors to generate a map of the space and odometry to create a path of the robot on the same map. At any timestep, the map and the path contain useful details about the environment and provide critical information to the exploration module, which is responsible for making decisions about the robot's next course of action, such as the next waypoint to move towards. With the next target location, the actuation module uses path planning techniques to command the actuators on the robot to physically move the robot to the desired target. The aforementioned pipeline is oblivious to the existence of other auxiliary sensors. For example, Infrared (IR) cameras typically have low resolution and small field of view, and as such collecting IR images collected by following a path suggested by depth sensor can result in low resolution imagery without sufficient spatial granularity. For example, in our experiments, the IR

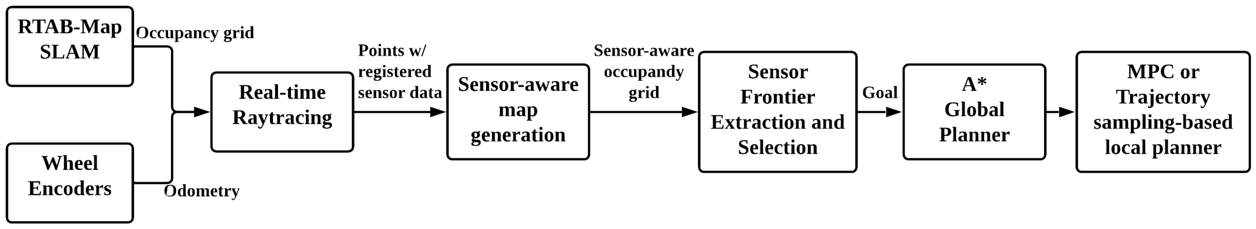

RTAB-Map SLAM
Wheel Encoders
Occupancy grid
Odometry
Real-time Raytracing
Points w/ registered sensor data
Sensor-aware map generation
Sensor-aware occupandy grid
Sensor Frontier Extraction and Selection
Goal
A* Global Planner
MPC or Trajectory sampling-based local planner

Figure 1: Overall framework of the system. The robot takes the map from the RTAB-Map SLAM algorithm and pose information from the wheel encoders, generating both classical frontiers and sensor-aware frontiers from which a goal point is selected. Then an A-star global planner and trajectory sampling-based local planner are employed to navigate the robot to the goal point.

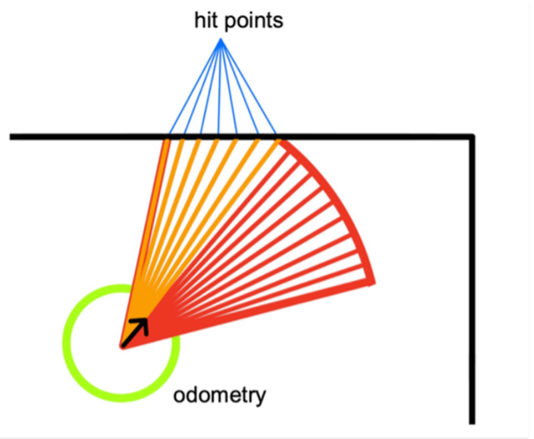


Figure 2: A visualization of our ray-tracing based approach. The green circle and arrow indicate the odometry of robot, and the orange lines represent the rays that hit obstacles.

camera and depth sensors have horizontal FOV of approximately 30 and 60 degrees respectively. Our proposed ray-tracing module aims to address the above problem by bridging the gap between data collection and SLAM. Specifically, it receives data from the SLAM module, incorporates the needs and constraints of auxiliary sensors into the map by extending rays from the robot path to nearby obstacles, and supplies the modified map with richer information and sensor-specific requirements to the exploration module to make better decisions about the next goal.

The overall block diagram of our pipeline is shown in Figure 1. For our implementation, we use RTAB-Map in the SLAM module and wheel encoders as odometry. We utilize the classical frontier exploration algorithm in the exploration module, which takes into account both distance and information gain when selecting the new frontier. Our algorithm takes the occupancy grid generated by depth sensor, and combines it with the robot odometry to generate the sensor-aware occupancy grid. The frontier goal selection algorithm, implemented within the explore lite library, then determines the goal location according to the sensor-aware occupancy grid. As such, we filer the map passed to the frontier goal selection to encode sensor coverage. For the actuation module, we use a combination of global A* planner and a local model predictive controller to navigate the robot to the desired locations.

We utilize ray tracing in a way similar to its application in computer graphics. A visualization of our raytracing approach is shown in Figure 2. Our algorithm takes two inputs from the

Algorithm 1 Real-time ray tracing

```
1:  Initialize Walls as $\{(x_1,y_1),...,(x_m,y_m)\}$ from wall pixels of
    occupancy grid
2:  Initialize Rays as empty list
3:  Initialize Wall Hits as empty list
4:  Initialize Odom as $\{(x_1,y_1,\theta_1),...,(x_n,y_n,\theta_n)\}$
5:  $l_{max} = 2$
6:  for $u_i = (x_i, y_i, \theta_i)$ do
7:    for $\Delta_\theta \in [-15, 15]$ do
8:      $l = 0.05$
9:      Initialize ray as empty list
10:     while $l < l_{max}$ do
11:       $x_i' = x_i + \cos(\theta_i + \Delta_\theta)l$
12:       $y_i' = y_i + \sin(\theta_i + \Delta_\theta)l$
13:       Append $(x_i', y_i')$ to ray
14:       if $(x_i', y_i') \in Walls$ then
15:         Append ray to Rays
16:         Append $(x_i', y_i')$ to Wall Hits
17:       end if
18:       $l += 0.05$
19:     end while
20:   end for
21: end for
22: return Wall Hits, Rays
```

SLAM module: a two-dimensional occupancy grid and the odometry data of the robot; the green circle with black arrow in the figure represents the odometry and the black lines are obstacles in part of the occupancy grid. Our proposed algorithm works as follows: we draw a ray, starting from the position of the robot on the occupancy grid. We then extend the ray in the orientation of the robot until it either hits an obstacle on the occupancy grid or reaches the maximum allowable distance. In our experiments, this is a predefined parameter set to 2 meters so as to encourage the robot to capture close-up thermal images to ensure detailed thermal images. If the ray hits an obstacle, we register the end of the ray as a “hit point”. We repeat this process for a certain angle range determined according to the valid FOV of the auxiliary sensor, which is 30 degrees in our case. The lines extending from the robot in the figure are visualizations of the rays, with red lines representing rays that do not hit obstacles and orange lines as rays that do hit obstacles. The pseudocode for our customized ray-tracing algorithm is shown below in Algorithm 1.

With the real-time ray tracing technique described above, we

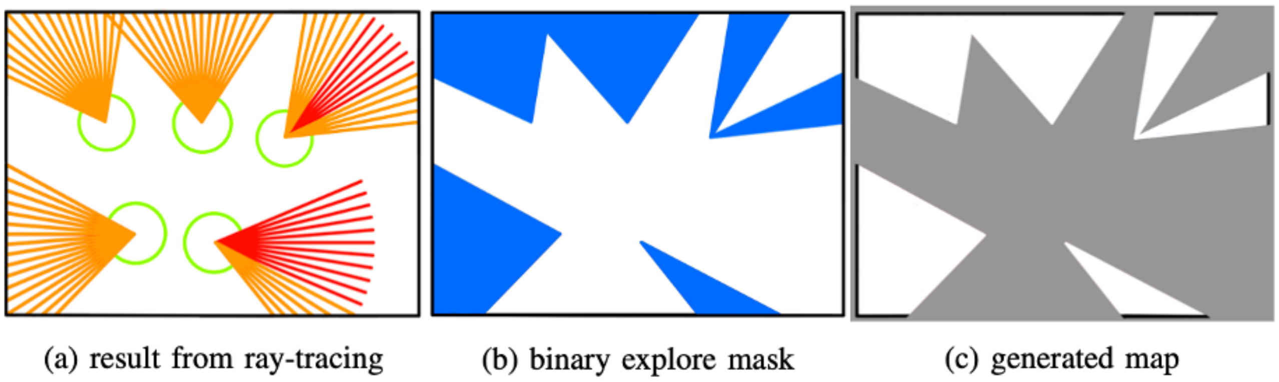


Figure 3: A visualization of using output from ray-tracing to construct a binary mask for occupancy grid, and superimposing over the original occupancy grid created by SLAM to compute the resulting map; (a) five odometries and their corresponding ray-tracing result; (b) the binary mask constructed by union of rays; (c) resulting occupancy grid after superimposing the binary mask over the original occupancy grid.

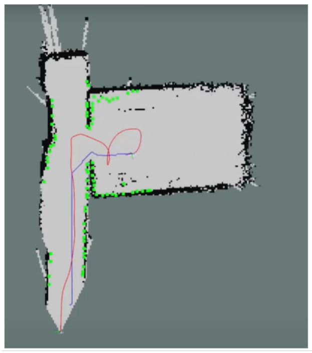


Figure 4: A demonstration of sensor coverage metric. The green dots indicate the locations with thermal data captured, and the black pixels indicate the walls, which are the locations required to have thermal data.

Algorithm 2 Sensor-aware occupancy grid generation from traced rays

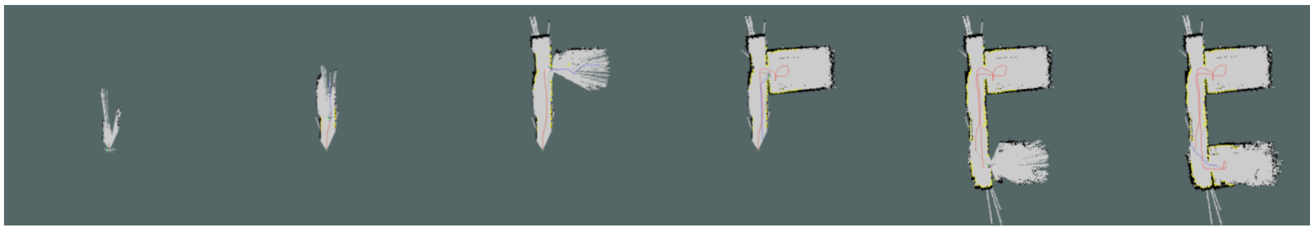
```
 1: Initialize True Map as occupancy gridmap generated from RTAB-Map
 2: Initialize Sensor-aware Map as empty map with same shape as True Map
 3: Initialize Rays as traced rays gemerated by real-time ray tracing
 4: Fill Sensor-aware Map with value representing unknown region (-1 in RTAB-Map setting)
 5: for $ray \in Rays$ do
 6:     for $point \in ray$ do
 7:         $x, y = point$
 8:         Sensor-aware Map[x, y] = True Map[x, y]
 9:     end for
10: end for
11: return Sensor-aware Map
```

can perform sensor-aware frontier exploration by constructing a masked map from the original occupancy grid map. We systematically underestimate the area that the robot has explored, to bias the robot to explore locations without auxiliary data. A visualization of this algorithm is shown in Figure 3. During runtime, after each iteration of ray tracing, we gather all "hit points" and their corresponding rays. For example, in the map shown in Figure 3(a), the green circles indicate the five odometries with valid sensor data captured, and the lines are visualization of ray tracing, with the orange lines indicating rays hitting obstacles. Next, we compute a union of these rays, mark the union area as explored, and construct a binary map of explored and unexplored accordingly. This is shown in Figure 3(b), with blue areas representing explored areas, which are computed by the union of orange lines in Figure 3(a). We superimpose the binary mask in Figure 3(b) on the occupancy grid, where unexplored positions on the mask translate to a value of unexplored in the corresponding positions on the occupancy grid. In the example given in Figure 3, the depth data collected at the five odometries is sufficient to construct the full occupancy map of the room. Superimposing the binary mask in Figure 3(b) on the occupancy grid results in the filtered map shown in Figure 3(c). By treating areas as unknown until valid auxiliary sensor data has been captured, we are incentivizing the robot to visit areas lacking sensor data coverage, therefore achieving higher sensor coverage compared to naive SLAM solution. The pseudocode for our map generation algorithm is shown below in Algorithm 2.

III. Experiments

We use the LoCoBot robot, which is a low- cost vehicle suitable for both exploration and manipulation. The LoCoBot's main components are a Kobuki base, an Intel NUC, an Intel Realsense D435 RGB-D camera, and a robotic arm. The above components were used to capture all the information required for mapping. In our experiments we used our proposed algorithm to map an unknown indoor space while collecting thermal images of the walls using a FLIR Boson 640.

We define a key metric, sensor coverage, to evaluate the quality of mapping and data collection. Specifically, sensor coverage is defined as the ratio of the area of locations with data to the area of locations required to have data. For our experiments, we require all walls to have thermal imagery data in order to thermally inspect the building envelope and to detect thermal anomalies. Figure 4 is an example generated by baseline method, which


(a) Naive baseline mapping experiment

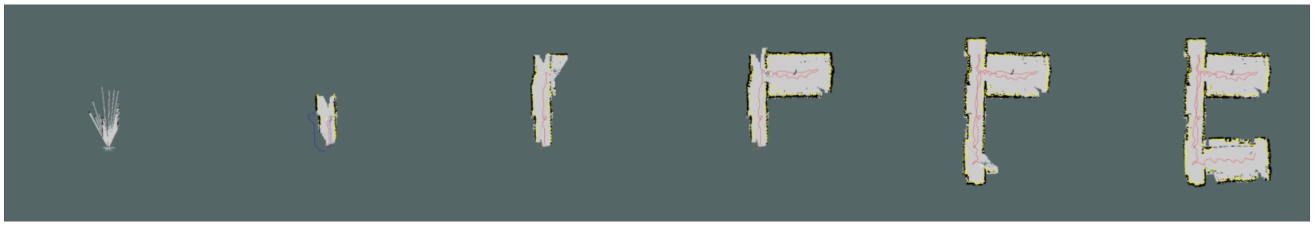

(b) Sensor-aware mapping experiment

Figure 5: Snapshots of two mapping experiments of the same environment in a commercial building setting: (a) baseline ; (b) proposed.

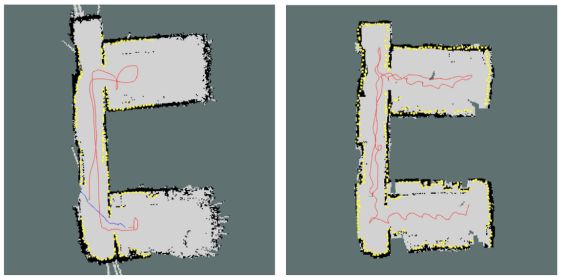

(a) baseline 49.0% coverage (b) sensor-aware 92.8% coverage

Figure 6: Maps generated during (a) baseline experiment (b) sensor-aware experiment. Yellow markers represent the locations with data collected. Red line represents the robot trajectory. Black markers represent the walls which are required to have data.

only uses a naïve SLAM algorithm. In Figure 4, the green dots indicate the locations with data captured, and the black pixels indicate the walls, which are the locations required to have data.

The first set of experiments are conducted in a commercial building setting with multiple rooms as shown in Figure 5. The baseline map is constructed with information from the RGB-D camera only. We can clearly see that the baseline method fails to cover the rooms with sufficient sensor data, while our proposed method successfully manages to collect data from all positions where thermal image data is needed. The visualization of the results of baseline and our proposed method is shown in Figure 6, which are the the zoomed in version of the right most portions of Figure 5. The baseline achieves a sensor coverage score of 49.0%. Our sensor-aware frontier exploration and mapping algorithm was tested in the same setting and achieves a sensor coverage score of 92.8%.

IV. Conclusions and Future Work

We have proposed a novel one pass sensor-aware exploration and mapping algorithm to simultaneously map an area and collect auxiliary sensor data. Thanks to the modular design, our pipeline of collecting thermal images using IR camera can be easily extended to other sensors by formulating new sensor constraints. For instance, to use gas sensors on the robot to collect gas dispersion in a buiding, we would need to replace ray tracing with a new sensor coverage model for gas dispersion. The concept of incorporating sensor requirements into exploration will also serve as an inspiration to more complex tasks. Compared to the traditional two-pass approach, our proposed one-pass method suits the need of robust auxiliary data collection while saving time and energy.

References

[1] B. Yamauchi, "A frontier-based approach for autonomous exploration," in Proceedings 1997 IEEE International Symposium on Computational Intelligence in Robotics and Automation CIRA'97. 'Towards New Computational Principles for Robotics and Automation', 1997, pp. 146–151.

[2] L. Lu, C. Redondo, and P. Campoy, "Optimal frontier-based autonomous exploration in unconstructed environment using rgb-d sensor," Sensors, vol. 20, no. 22, p. 6507, 2020.

[3] D. Delling, P. Sanders, D. Schultes, and D. Wagner, "Engineering route planning algorithms," in Algorithmics of large and complex networks. Springer, 2009, pp. 117–139.

[4] X. Li, Z. Sun, A. Kurt, and Q. Zhu, "A sampling-based local trajectory planner for autonomous driving along a reference path," in 2014 IEEE intelligent vehicles symposium proceedings. IEEE, 2014, pp. 376–381.

[5] J. Fuentes-Pacheco, J. Ascencio, and J. Rendon-Mancha, "Visual simultaneous localization and mapping: A survey," Artificial Intelligence Review, vol. 43, 11 2015.

[6] O. Kähler, V. A. Prisacariu, and D. W. Murray, "Real-time large-scale dense 3d reconstruction with loop closure," in Computer Vision – ECCV 2016, B. Leibe, J. Matas, N. Sebe, and M. Welling, Eds. Cham: Springer International Publishing, 2016, pp. 500–516.

[7] M. Labbe and F. Michaud, "Appearance-based loop closure detection for online large-scale and long-term operation," Robotics, IEEE Trans- actions on, vol. 29, pp. 734–745, 06 2013.

[8] "explore lite," accessed: 2021-05-09. [Online]. Available: http://wiki.ros.org/explore lite

[9] A. Murali, T. Chen, K. V. Alwala, D. Gandhi, L. Pinto, S. Gupta, and A. Gupta, "Pyrobot: An open-source robotics framework for research and benchmarking," arXiv preprint arXiv:1906.08236, 2019.

[10] "rtabmap ros." [Online]. Available: http://wiki.ros.org/rtabmap ros

[11] "rrt exploration." [Online]. Available: http://wiki.ros.org/rrt exploration

[12] X.Li,Z.Sun,A.Kurt,andQ.Zhu,"Asampling-basedlocaltrajectory planner for autonomous driving along a reference path," in 2014 IEEE intelligent vehicles symposium proceedings. IEEE, 2014, pp. 376– 381.

[13] M. Labbe and F. Michaud, "Rtab-map as an open-source lidar and visual simultaneous localization and mapping library for large-scale and long-term online operation," Journal of Field Robotics, vol. 36, no. 2, pp. 416–446, 2019.

[14] P. Hart, N. Nilsson, and B. Raphael, "A formal basis for the heuristic determination of minimum cost paths," IEEE Transactions on Systems Science and Cybernetics, vol. 4, no. 2, pp. 100–107, 1968. [Online]. Available: https://doi.org/10.1109/tssc.1968.300136

[15] M. Labbe and F. Michaud, "Online global loop closure detection for large-scale multi-session graph-based slam," in 2014 IEEE/RSJ International Conference on Intelligent Robots and Systems, 2014, pp. 2661–2666.

[16] Stanford Artificial Intelligence Laboratory et al., "Robotic operating system." [Online]. Available: https://www.ros.org

Author Biography

Zixian Zang is currently a senior undergraduate student at UC Berkeley. His work focuses on SLAM and exploration, as well as machine learning and computer vision.

Haotian Shen is currently a junior undergraduate student at UC Berkeley. His work focuses on systems and legged robotics.

Lizhi Yang is currently a senior undergraduate student at UC Berkeley. His work focuses on legged robotics, human robot interaction, and computer vision.

Avideh Zakhor is currently Qualcomm Chair and professor in EECS at U.C. Berkeley. Her areas of interest include theories and applications of signal, image and video processing and 3D computer vision. She was selected as Electronic Imaging scientist of the year by SPIE, is a IEEE fellow, and has received a number of best paper awards from IEEE societies of signal processing, circuits and systems, and solid state circuits.