Proc. 2015 IEEE/SICE Int. Symp. on System Integration (SII-2015), pp. 599-604, Nagoya, Japan, Dec. 2015.

Exploration and Observation Planning for 3D Indoor Mapping

Yuki Okada and Jun Miura
Department of Computer Science and Engineering
Toyohashi University of Technology

***Abstract*— This paper deals with an observation planning for indoor 3D mapping. We consider the case where the robot makes a map with different resolutions; it observes informative regions from near positions. Since observing a region with a higher resolution requires more time, we need to carefully choose viewpoints for such observations. This paper therefore focuses on viewpoint planning for observing a set of informative (or important) regions. We develop a robot system which first explores an environment to make a 2D map using a 2D LIDAR and then uses the map for localizing informative regions and plans a sequence of viewpoints from which all of the regions can be observed with a RGB-D camera. The viewpoint planner tries to minimize the sum of view and travelling cost. We also investigate the trade-off between planning cost and plan quality.**

I. Introduction

Mapping is one of the fundamental functions of mobile robots. They explore environments, observe at various positions, and integrate them to make the description of environments (maps). This paper deals with 3D mapping of indoor environments, especially a viewpoint planning for efficient mapping.

Mapping by mobile robots is usually handled by SLAM (simultaneous localization and mapping) methods [1]. To make a 3D map of environment, various SLAM methods have been proposed for indoor (e.g., [2]) and for outdoor (e.g., [3]). These works mainly focus on how to obtain precise maps of the environment from a given sequence of observations.

Making a map of an unknown environment requires a robot to explore the environment autonomously. Several exploration strategies such as frontier-based [4] and integrated exploration [5] have been proposed. Exploration is usually for covering a whole environment and is not necessarily intended for finding specific objects or for an uneven environment mapping which changes resolution for various places.

Object search is also a viewpoint planning problem. Shubina and Tsotsos [6] and Saidi et al. [7], for example, propose viewpoint planning methods based on statistical inference. Aydemir et al. [8] utilize high-level knowledge on spatial relations between objects to guide a search behavior. Masuzawa and Miura [9] formulated a combined exploration and object search method. These works are for developing algorithms for efficient object search.

We are interested in constructing a 3D map of the environment which has different resolutions depending on the importance of the region. A region with relatively small objects such as PCs and books is described with a high resolution for reliable object recognition using, for example KinectFusion [10], while a low resolution description is enough for regions only with walls or large tables without any objects. Since observing a region with a higher resolution requires more time, we need to carefully choose viewpoints for such observations. This paper therefore focuses on viewpoint planning for observing a set of important (or informative) regions.

Choosing a set of viewpoints which cover designated regions has been dealt with in several ways. Art Gallery Problem (AGP) [11] is a problem of determining a minimum number of guards such that they cover the whole interior regions of an environment. A sensor coverage problem (e.g., [12]) in robotic context is a variant of AGP.

Watchman Route Problem (WRP) is similar to AGP but it tries to find shortest routes from which every point in a given space is visible [13]. This could be solved by a combination of AGP and Traveling Salesman Problem (TSP). WRP is applied to various inspection planning or sensor placement problem [14], [15].

In this paper, we deal with a problem of 3D mapping by a mobile robot in two steps. In the first step, the robot explores an unknown environment and constructs a 2D map. In the second step, the robot localizes potentially informative regions in the 2D map, and generates a sequence of viewpoints which collectively cover all such regions. The viewpoint planner in the second step tries to minimize the sum of observation and traveling cost. Since planning algorithms can be evaluated in terms of not only the quality of generated plans but also how fast plans are generated, we investigate the trade-off between planning cost and plan quality. Such a trade-off has little been considered so far in vision planning context. Note that currently we run a 2D exploration and a 3D mapping sequentially because usual 2D LIDARs have a much longer measurable range than RGB-D cameras. Integration them for a more efficient mapping is future work.

The rest of the paper is organized as follows. Sec. II explains our robot system and its software configuration. Sec. III describes an exploration strategy with some result. Sec. IV explain the viewpoint planning algorithm in detail. Sec. V shows an experimental result of autonomous 3D mapping. Sec. VI investigates the trade-off between the cost of viewpoint planning and the quality of generated viewpoint sequence in an experimental setting. Sec. VII concludes the paper and discusses future work.

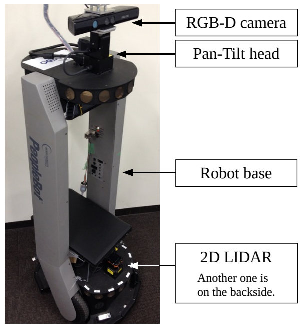


Fig. 1. The robot for 3D mapping.

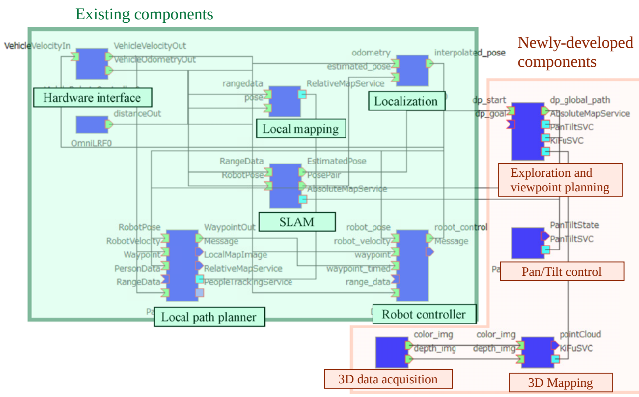


Fig. 2. Configuration of RT components of the robot system. Each blue box is an RT-component. The green group indicates already-developed RT components, while the red group indicates those developed in this research.

II. System Overview

A. Hardware

Fig. 1 shows the robot. It is a PeopleBot by Mobile Robots equipped with two LRFs (Hokuyo UTM-30LX) and a Kinect with a Pan-Tilt head (FLIR PTU-D46). The LRFs are used for exploration and 2D mapping, while Kinect is used for point cloud acquisition.

B. Basic Software

All software modules are implemented as RT Components (RTCs), running in the RT-Middleware environment [16]. Fig. 2 shows the configuration of the RT Components. We use the following existing (developed either by ourselves or by others) RTCs.

- SLAM and MCL (Monte Carlo Localization) RTC: this RTC uses routines from MRPT (Mobile Robot Programming Toolkit) [17].
- Motion planing RTCs: these RTCs calculate a collision-free path based on a randomized search with considering a bias from a potential function called Arrival Time Field [18].
- Local mapping RTC: this performs a usual update on a probabilistic occupancy map around the robot.
- RTCs for controlling the robot and the sensors.

III. 2D Mapping by Automatic Exploration

The input to the viewpoint planning treated in this paper is a 2D map of the environment. Since we deal with mapping of an unknown environment, A 2D map must also be constructed automatically. We here describe our exploration algorithm, which is basically frontier-based [4] with an information gain-based thresholding.

Each cell of a probabilistic occupancy map is labeled as one of the following: free, unknown, and occupied. Two thresholds are given in advance for this classification. A frontier point is a point which is free and adjacent to an unknown point. All frontier points are partitioned into clusters with a distance threshold th_d (currently, $th_d = 0.5\,m$).

Each cluster c has a representative point $\boldsymbol{x}_c$, which is the frontier point closest to the centroid of all points. This point is used as the viewpoint if this is a frontier and if the robot observes at this frontier. We also define a region $\Omega(\boldsymbol{x}_c)$ which is observable from $\boldsymbol{x}_c$. This region is calculated considering the sensor range and predicted visibility constraints.

Now we give a definition of frontier. Cluster c is considered as a frontier iff:

- The number of its frontier points is more than a threshold th_n (currently, $th_n = 13$), and
- The information gain by observation at $\boldsymbol{x}_c$ is large enough.

Assuming that a complete sensing which let the entropy to zero, the information gain is equal to the current entropy $H(c)$ of the observable region $\Omega(\boldsymbol{x}_c)$:

$$H(c) = -\sum_{\boldsymbol{x} \in \Omega(\boldsymbol{x}_c)} P(\boldsymbol{x}) \log P(\boldsymbol{x}) \qquad (1)$$

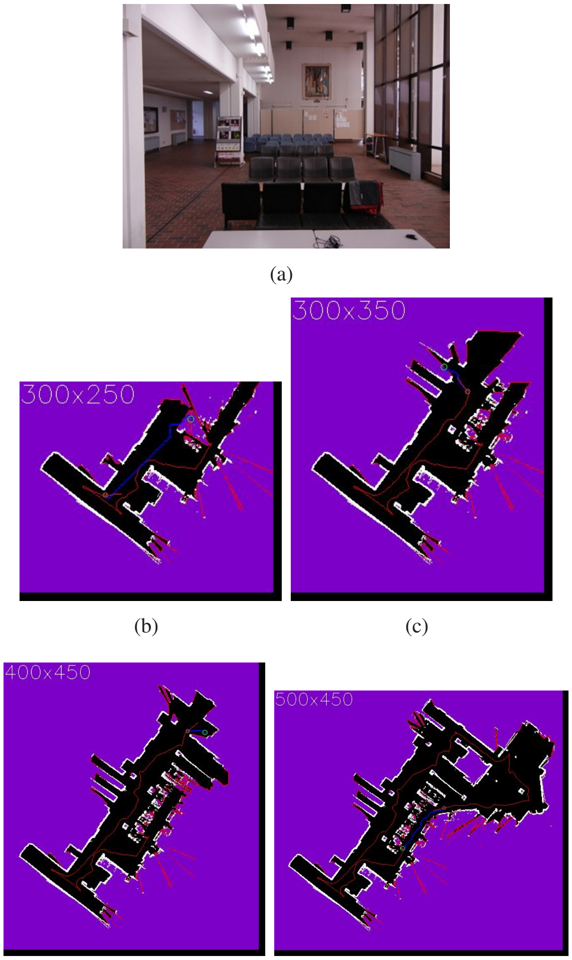
Now we have a set of frontiers. González-Baños and Latombe [19], for example, use a utility function which combines the information gain and the distance to the frontier. This could be useful when the robot is searching for some interesting places. Since our objective is to make a whole map of the environment, the robot has to eventually visit all places. We thus use a simple utility function considering only the distance; this makes the robot take the closest-frontier strategy [4]. Fig. III shows an example exploration for 2D mapping of an unknown place.

IV. Viewpoint Planning

The objective of viewpoint planning is to obtain a sequence of viewpoints by which a 3D description of the scene is efficiently constructed. We first define the regions to observe (or informative regions), and then describe a planning algorithm.

A. Informative regions

Maps usually contain geometric and semantic information. The former describes the shape of 2D or 3D free space and obstacles, while the latter describes many kinds of information which is useful for task execution and/or human-robot communication, such as object placements and scene class (living room, kitchen, ...). In this paper, we are interested in



(a)

(b) (c)

(d) (e)

Fig. 3. Automatic exploration example. (b)∼(e) show process of exploration and incremental mapping. (a) is a view of the central part of the final map (e), from the bottom side.

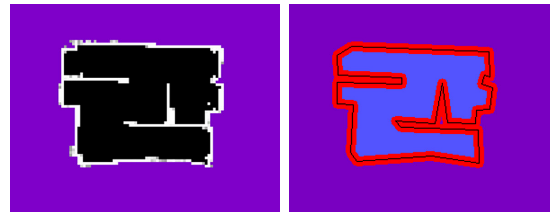
acquire 3D data for the latter, based on 2D map obtained by SLAM.

We currently use a simple heuristic for determining informative regions. We see in a usual room useful objects (in the shelves) near walls or on desks and tables. Assuming that outlines of shelves and desks/tables can be detected as boundaries of obstacle regions, such objects will lie within a certain distance to the boundaries.

Based on the above heuristic, we divide a 2D map into four regions:

- Informative region: regions to observe near object boundaries.
- Boundary region: object boundaries, the orientations of which are used for assessing incident constraints.
- Free space region: region where the robot can move without collision. Viewpoints are selected in this region.
- Other region: this is not maneuverable nor necessary to observe.

The viewpoint planning generates a sequence of viewpoints in the free space regions such that observations there collectively cover all informative regions. This region classification is done after a polygonal approximation of obstacle boundaries. Fig. 4 shows an example result of region classification from a probabilistic occupancy map. Since the method is grid-based, we can easily add other heuristics for assessing the informativeness of regions.



(a) Input 2D map. (b) A classified map.

Fig. 4. An example result of region classification. Red, black, blue, and purple regions in the classified map indicate informative, boundary, free space, and other regions, respectively.

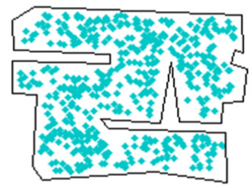


Fig. 5. Randomly placed viewpoint candidates.

B. Viewpoint sequence generation overview

A viewpoint sequence should satisfy the following conditions:

- All viewpoints are safe, that is, they are in the free space region with some safety margin.
- Observations there collectively covers all informative regions.
- The cost of the observation actions should be low enough.

Since we are interested in minimizing the total of the view and travelling cost [20], and since the cost of optimization is usually huge, we take a randomized approach, namely, we generate a set of viewpoints with satisfying the coverage condition randomly and calculate the total cost. We repeat this process for a certain times, and takes the best sequence generated so far. The following subsections explain the actual procedures in detail.

C. Viewpoint generation

We adopt randomized AGP [12] for viewpoint generation. This iteratively puts a viewpoint and updates the coverage until the total coverage exceeds a threshold. We apply this method to our robot system. The detailed steps are as follows.

1) Viewpoint candidate generation: A certain number of viewpoint candidates are randomly generated in the free spaces. Fig. 5 shows an example of initial candidate generation in the free space shown in Fig. 4(b). For each candidate, the size of informative regions observed from this candidate is then calculated as its weight. The range and the incident limitation are considered in this calculation.

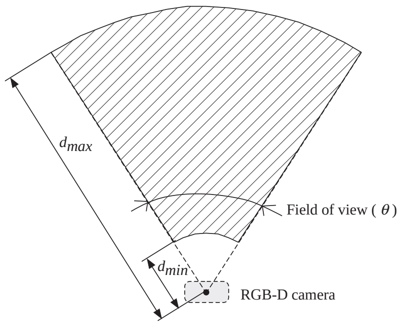


Fig. 6. Observed region of Kinect sensor. d_{min} and d_{max} are $0.5\,m$ and $8.0\,m$, respectively. θ is $57\,deg$.

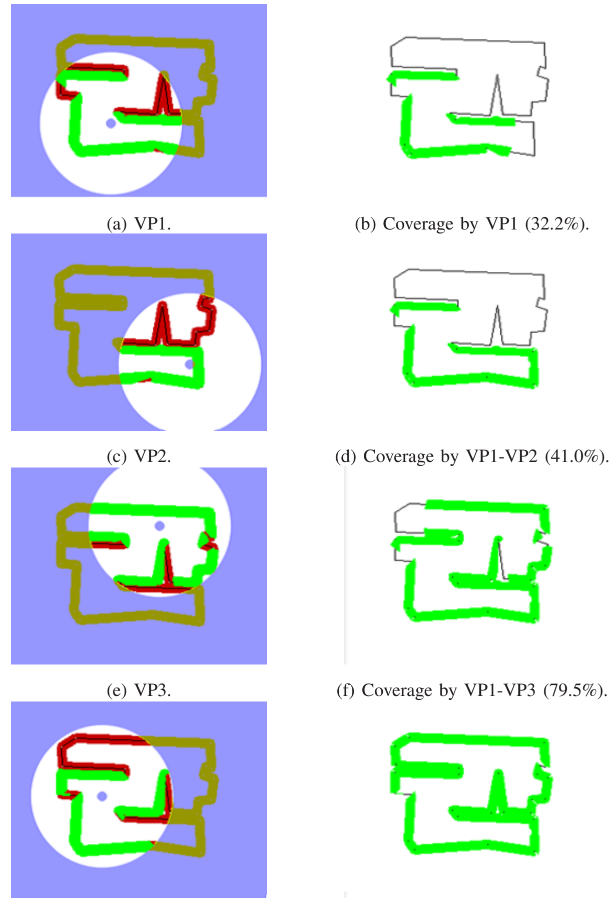
Fig. 6 shows the observed region of Kinect sensor. The range limitation is set to between $0.5\,m$ and $8.0\,m$. The incident angle is calculated as the difference between the normal at an observed point and the viewing direction. For a point other than boundary points, the normal at the nearest boundary point is used. The incident limitation is empirically set to $80\,deg$. Example calculations of covered area considering the two limitations are shown in the left column of Fig. 7.

2) Probabilistic sampling of viewpoint set: Viewpoints are sampled iteratively based on their probability, which are normalized weight values, and the accumulated coverage is updated. This iteration stops when the total coverage exceeds a threshold (currently, 95%). To get a high coverage with keeping the number of viewpoints small, a newly sampled viewpoint is added to the set only when the coverage increase by it is more than a certain value (currently, 10%). Fig. 7 shows an example step of iteratively selecting viewpoints (VPs).

D. Cost calculation

A plan for traveling and observing at viewpoints is generated for each set. The cost of a plan is the sum of the observation and the traveling cost. The robot observes and obtain point cloud data of the nearby informative regions by rotating the RGB-D camera. The time of observation is approximately calculated as the time of rotation, which is determined from the field of view (FOV) of the camera ($57\,deg$; see Fig. 6) and the orientation range the robot should cover at that viewpoint. We have here a simplified assumption that an observation at a viewpoint is independent of those in the others. Based on this assumption, the necessary amount time of rotation is given by the division of the rotational range by the FOV.

Concerning the traveling cost, the order of visiting viewpoints is crucial. We thus need to solve a traveling salesman problem (TSP). We use a simple 2-opt method [21] for this TSP. This algorithm randomly picks up two pairs of consecutive viewpoints, and switches the pairs if that makes the total traveling distance shorter. The distance between viewpoints is calculated by applying A* algorithm in the occupancy map, and then is divided by an average robot speed to calculate the traveling time. Fig. 8 shows the initial and the final route for traveling four viewpoints.



(a) VP1. (b) Coverage by VP1 (32.2%).
(c) VP2. (d) Coverage by VP1-VP2 (41.0%).
(e) VP3. (f) Coverage by VP1-VP3 (79.5%).
(g) VP7. (h) Coverage by VP1-VP7 (96.5%).

Fig. 7. Iterative selection of viewpoints. This case requires seven viewpoints to cover more than 95% of the whole informative regions. Respective coverage and accumulated coverage are shown for VP1, VP2, VP3, and VP7, respectively.

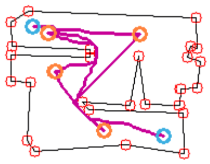
The cost C_e of executing a plan with viewpoints $v_i\,(i = 0, \cdots, n{+}1)$ (v_0 and v_{n+1} are the start and the goal position, respectively) is given by:

$$C_e = \sum_{i=0}^{n} \frac{dist(v_i, v_{i+1})}{V} + \sum_{i=1}^{n} C_o(v_i),$$

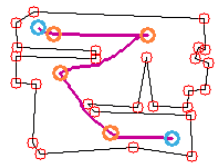
where $dist(v, v')$ is the length of optimal path connecting two positions, V is the average robot speed, and $C_o(v)$ is the time of observation at viewpoint v, which depends on the range of sensor orientation changes.

E. Choosing the best plan

Randomized algorithms may produce inefficient plans. We therefore perform the viewpoint sequence generation for a certain times, and choose the best plan. In general, as the number of trials increases, the quality of the plan will increases, but at the same time, planning cost will also increase. The best number of trials should be determined considering this trade-off. We discuss about this trade-off in Sec. VI.



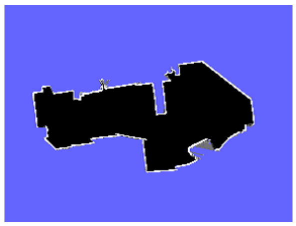
(a) Initial route.



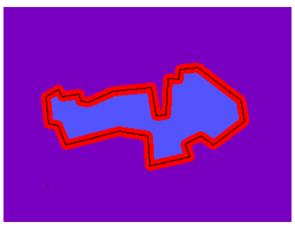
(b) Final route.

Fig. 8. A TSP result. Blue circles indicate the start and the goal position; four orange circles indicate viewpoints.

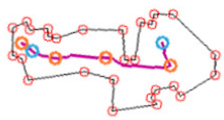
(a) A room scene.



(b) Probabilistic occupancy map.



(c) Classified map.



(d) Planning result.

Fig. 9. Experimental scene, 2D exploration, and viewpoint planning. In (d), Blue circles indicate the start and the goal position; four orange circles indicate viewpoints.

V. Experimental Result

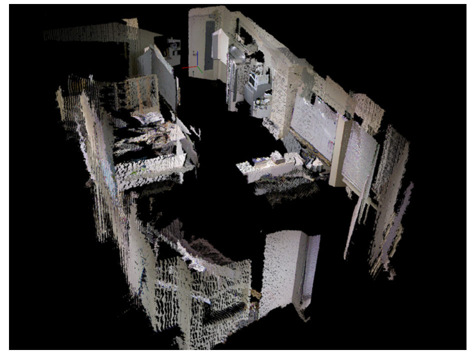
This section describes an integrated experiment in a room environment. Fig. 9(a) shows the room scene, and Fig. 9(b) is the 2D map the robot obtained by observations at four frontier points. Fig. 9(c) is the classified map. The robot generates a plan which covers a certain portion of informative regions (red regions) while moving inside the free space region (blue regions). Fig. 9(d) is the viewpoint planning result. Four viewpoints are enough for this environment.

A 3D map is currently represented by a set of 3D points with color. The point cloud data obtained at the viewpoints are transformed in a world coordinate frame using the localization result. Fig. 10 shows the obtained 3D map from two different virtual viewing positions.

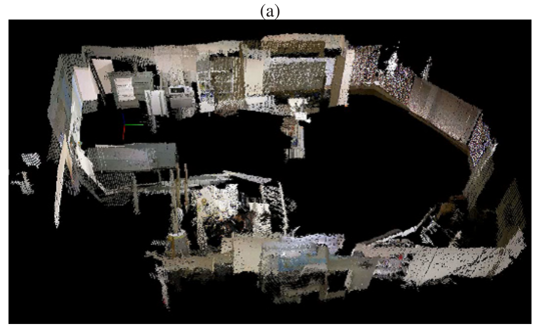
VI. Trade-off Analysis

There is a trade-off between planning cost and plan quality. As mentioned above, randomized algorithms can usually generate better results as the number of trials increases, but the planning cost also increases. Therefore, it is not desirable to try too many times for obtaining a near-optimal plan.

One possible criterion to determine the *time for planning* (i.e., the number of trials in our problem) is to minimize the sum of the planning cost C_p and the execution cost C_e. Anytime algorithm-based meta-planning [22] could be



(a)



(b)

Fig. 10. Obtained 3D map from two different virtual viewing positions.

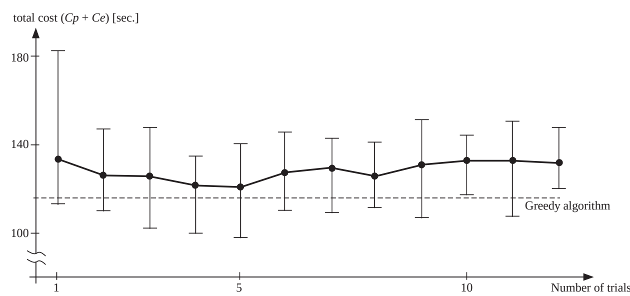


Fig. 11. Planning cost versus plan quality trade-off. Bars indicate the maximum and the minimum values.

one approach to determining the optimal time for planning. It is, however, not always the case where we can have a procedure for explicitly doing the meta-planning. In this paper, therefore, as a first step, we analyze the relationship between the number of trials N and the total cost $(C_p + C_e)$.

For a value of N, we calculate N potential plans and take the minimum cost one as described in Sec. IV-E. For a statistical analysis, we repeat this 30 times for each N and calculate the average, the minimum, and the maximum value. The results are summarized in Fig. 11. We can clearly see the trade-off; that is, too small and too large N's are not desirable. $N = 5$ is the best for this problem.

The horizontal line in the figure is the planning and the execution cost for a greedy algorithm. The algorithm calculates the plan in the following steps:

1) Put viewpoint candidates randomly in the free space.
2) Calculate the area coverage of each candidate and sort them in the descending order of the area coverage.
3) Pick up the candidate with the largest coverage.

4) Repeat steps 2) and 3) until the accumulated coverage exceeds a threshold (95% is used as the threshold).
5) Calculate the best travel for the obtained set of viewpoints and calculate the total cost.

Compared in the average cost, the greedy algorithm is slightly better than the randomized algorithm. We need to further investigate this trade-off relationship for various environment and parameter settings.

VII. Conclusions and Discussion

This paper describes a 3D mapping system which combines exploration and observation planning. The system first explores an unknown environment to make a 2D map. The map is then analyzed to locate informative regions where further observations are needed. The system solves a viewpoint planning problem which is to generate a sequence of safe viewpoints from which all of the regions can be observed. Since obtaining the optimal sequence is intractable, we employ a randomized solution. We also investigate the trade-off between the planning cost (i.e., the number of trials for viewpoint sequence generation) and the plan quality, which is the sum of observation and traveling cost.

The presented work is a first step toward more efficient and versatile 3D mapping, and several future research directions are possible:

a) Integrated 2D and 3D mapping: Since we focus on the viewpoint planning in this paper, the 2D and the 3D mapping are separated, and the result of the former is used as input to the latter. Integrating them could improve the efficiency of mapping, but at the same time, the solution to viewpoint planning problem becomes incremental and could be more difficult to optimize. In such an integrated mapping, it is interesting to determine informative regions in 3D, which could realize a more efficient 3D mapping.

b) Adaptive resolution 3D mapping and object recognition: The constructed 3D map is at present just a collection of observations (point clouds) at planned viewpoints. It would be necessary in some applications to adaptively control the resolution of mapping depending on some importance of regions. This will require to consider more constraints such as a variable range constraint in the viewpoint planning. It is also desirable to recognize objects in the scene and record them in the map. This could additionally require a local observation planning for object recognition (e.g., [9]).

c) On-line control of planning time: From the meta-planning perspectives, it is interesting to investigate an on-line control of planning time in mapping, especially under a time limitation. This might require an advance assessment of the trade-off between the planning cost and the plan quality based on a model such as performance profile [23].

Acknowledgement

The authors would like to thank Dr. Matthieu Boussard for developing an earlier version of exploration algorithm and Dr. Igi Ardiyanto for supporting a part of the experiments. This work is in part supported by JSPS KAKENHI Grant Number 25280093.

References

[1] S. Thrun, W. Burgard, and D. Fox. *Probabilistic Robotics*. The MIT Press, 2005.

[2] F. Endres, J. Hess, N. Engelhard, J. Strum, D. Cremers, and W. Burgard. An Evaluation of the RGB-D SLAM System. In *Proceedings of 2012 IEEE Int. Conf. on Robotics and Automation*, pp. 1691–1696, 2012.

[3] A. Nüchter, K. Lingemann, J. Hertzberg, and H. Surmann. 6D SLAM – 3D Mapping Outdoor Environments. *J. of Field Robotics*, Vol. 24, No. 8–9, pp. 699–722, 2007.

[4] B. Yamauchi. A Frontier-Based Approach for Autonomous Navigation. In *Proceedings of the 1997 IEEE Int. Conf. on Computational Intelligence in Robotics and Automation*, pp. 146–151, 1997.

[5] A.A. Makarenko, S.B. Williams, F. Bourgault, and H.F. Durrant-Whyte. An Experiment in Integrated Exploration. In *Proceedings of 2002 IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*, pp. 534–539, 2002.

[6] K. Shubina and J.K. Tsotsos. Visual Search for an Object in a 3D Environment Uing a Mobile Robot. *Computer Vision and Image Understanding*, Vol. 114, pp. 535–547, 2010.

[7] F. Saidi, O. Stasse, K. Yokoi, and F. Kanehiro. Online Object Search with a Humanoid Robot. In *Proceedings of 2007 IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*, pp. 1677–1682, 2007.

[8] A. Aydemir, K. Sjöö, J. Folkesson, A. Pronobis, and P. Jensfelt. Search in the Real World: Active Visual Object Search Based on Spatial Relations. In *Proceedings of 2011 IEEE Int. Conf. on Robotics and Automation*, pp. 2818–2824, 2011.

[9] H. Masuzawa and J. Miura. Observation Planning for Environment Information Summarization with Deadlines. In *Proceedings of 2010 IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*, pp. 30–36, 2010.

[10] R.A. Newcombe, S. Izadi, O. Hilliges, and D. Molyneaux. KinectFusion: Real-Time Dense Surface Mapping and Tracking. In *Proceedings of 10th IEEE Int. Symp. on Mixed and Augmented Reality*, pp. 127–136, 2011.

[11] J. O'Rourke. *Art Gallery Theorems and Algorithms*. Oxford University Press, 1987.

[12] H. González-Baños and J.-C. Latombe. A Randomized Art-Gallery Algorithm for Sensor Placement. In *Proceedings of 7th Annual Symp. on Computational Geometry*, pp. 232–240, 2001.

[13] W.-P. Chin and S. Ntafos. Optimum Watchman Route. In *Proceedings of 2nd Annual Symp. on Computational Geometry*, pp. 24–33, 1986.

[14] J. Faigl, M. Kulich, and L. Preucil. A Sensor Placement Algorithm for a Mobile Robot Inspection Planning. *J. Intelligent Robot Systems*, Vol. 62, pp. 329–353, 2011.

[15] T. Danner and L. Kavraki. Randomized Planning for Short Inspection Paths. In *Proceedings of 2000 IEEE Int. Conf. on Robotics and Automation*, pp. 971–976, 2000.

[16] N. Ando, T. Suehiro, and T. Kotoku. A Software Platform for Component Based RT System Development: OpenRTM-aist. In *Proceedings of the 1st Int. Conf. on Simulation, Modeling, and Programming for Autonomous Robots (SIMPAR '08)*, pp. 87–98, 2008.

[17] MRPT. `http://www.mrpt.org/`.

[18] I. Ardiyanto and J. Miura. Real-Time Navigation using Randomized Kinodynamic Planning with Arrival Time Field. *Robotics and Autonomous Systems*, Vol. 60, No. 12, pp. 1579–1591, 2012.

[19] H. González-Baños and J.-C. Latombe. Navigation Strategies for Exploring Indoor Environments. *Int. J. of Robotics Research*, Vol. 21, No. 10-11, pp. 829–848, 2002.

[20] P. Wang, R. Krishnamurti, and K. Gupta. View Planning Problem with Combined View and Traveling Cost. In *Proceedings of 2007 IEEE Int. Conf. on Robotics and Automation*, pp. 711–716, 2007.

[21] O. Mersmann, B. Bischl, J. Bossek, H. Trautmann, M. Wagner, and F. Neumann. Local Search and the Traveling Salesman Problem: A Feature-Based Characterization of Problem Hardness. In *Learning and Intelligent Optimization*, Vol. LNCS 7219, pp. 115–129. 2012.

[22] M. Boddy and T. Dean. Solving Time-Dependent Planning Problems. In *Proceedings of IJCAI-89*, pp. 979–984, 1989.

[23] T. Dean and M. Boddy. An Analysis of Time-Dependent Planning. In *Proceedings of AAAI-88*, pp. 49–54, 1988.