Localizability-based Topological Local Object Occupancy Map for Homing Navigation

Hwiyeon Yoo[1] and Songhwai Oh[1]

***Abstract*— In this paper, we proposed a localizability-based topological local object occupancy map (TLO2M) for homing navigation. The proposed approach is a combination of topological and metric map representations. We utilize object detection to advance the occupancy grid map and train a structural localizability measuring network with it. As a result, the TLO2M is built based on structural localizability and feature similarity. The proposed method shows a 0.955 success rate of the homing task at the Gibson environments.**

I. INTRODUCTION

Homing is a navigation task which is a problem of returning to the agent's initial point by following the reverse path of the agent came. One strategy to solve this problem is to build an environment map during the first journey, and use it to follow back to the starting point. At this point, the further approaches are diverse based on the way of map representations. One can build metric maps using simultaneous localization and mapping (SLAM) algorithms [3], [8], or topological graph maps [7], [9], or even internal memory maps [6]. Each of these strategies have advantages and disadvantages. The metric maps using SLAM algorithm are easy to use when the map is built completely, but since the SLAM algorithms need an exploring stage for mapping, the map built on the online agent trajectory is not reliable. In addition, the SLAM-based algorithms require high input frequency and do not contain semantic information of the environment. On the other hand, topological graph maps are usually built based on semantic information such as similarity [9], and reachability [7]. However, navigation between the nodes of the topological graph maps is less reliable since the map does not contain dense information of free space and obstacles. In the case of internal memory [6], it is an efficient way to keep information of the online trajectory, but the memory is not robust when the agent is out of the exact trajectory memorized.

In this paper, we propose a novel visual path map representation called topological local object occupancy map (TLO2M) which is a combination of the topological and metric map representations. To build a TLO2M, we utilize object class information and the structural uniqueness of a location. With the TLO2M representation, we build an efficient map for the homing navigation that contains both semantic information and free space information.

[1]H. Yoo, and S. Oh are with the Department of Electrical and Computer Engineering and ASRI, Seoul National University, Seoul 08826, Korea hwiyeon.yoo@rllab.snu.ac.kr, songhwai@snu.ac.kr

II. RELATED WORK

Kumar et al. [6] proposed the importance of the path-following and homing problems in their paper. They tackled the problem by building intrinsic visual memory of the first trajectory that can be used in the path-following or homing phases. Their approach showed good results on both path-following and homing navigation in simple trajectories under 40 steps. However, when the length of the trajectory grows, the performance of both tasks drops significantly which shows the weakness of the proposed method in long-term scenarios. Also, the visual memory proposed in [6] does not contain extrinsic information, it is not robust enough when the agent is out of the original trajectory.

Chaplot et al. [1] used object detection to make a grid map with semantic information. This is similar to our approach of making object occupancy map which will be described in Section III-B. In this paper, we used a probabilistic approach based on the occupancy grid map rather than the learning-based denoising approach which was used in [1].

III. METHOD

A. Overview

In this paper, we tackle a homing navigation, which is a problem of returning to the initial point by following the reverse path of the agent came. We assume that the agent only has an RGB-depth (RGB-D) camera sensor with a low input frequency. To solve the homing navigation in this condition, we propose a novel visual path memory called the topological local object occupancy map (TLO2M). The TLO2M is a graph based map representation, where the nodes are local map of representative locations $\mathbf{N} = \{n_1, ..., n_k\}$ among the agent's original trajectory $\mathbf{T} = \{t_1, \ldots, t_n\}$. In this section, we describe steps to build the TLO2M and the homing navigation method using the TLO2M.

B. Object Occupancy Map

For the nodes of TLO2M, we use a metric map representation of the local environment based on the occupancy grid map. The occupancy grid map $M_{occ} \in \mathbb{R}^{m \times m}$ is a map that represents the probability of the existence of obstacles at each cell. A 2D occupancy grid map can be built by projecting 3D point clouds of observed obstacles over time. With a depth camera and the camera intrinsic parameters, we can obtain 3D point clouds by back-projecting the depth image (D_t) at time step t. However, the conventional occupancy grid map only contains information on the location of obstacles and has a limitation of describing semantic features of the environment. In this paper, we proposed an object occupancy

map (O2M), $L \in \mathbb{R}^{m \times m \times (1+c)}$, which is an advanced version of the occupancy grid map that can represent the category of objects as well as locations. The O2M has extra c channels for each grid that corresponds to c kinds of object categories. Similar to the ordinary occupancy grid map, each c channel represents the probability of the existence of the corresponding object at the target location. To get the object category information, we use the pretrained Mask R-CNN [4] on the RGB observations, which outputs object masks and their probability. Since the object masks on an RGB image can be projected to the same location as the corresponding depth image, we can give the point clouds obtained from the depth image the object class information. The O2M is built by projecting these point clouds containing object class information on a 2D plane. Since the O2M is the sum of point clouds across the height, multiple channels among c classes can be occupied at a single cell of an O2M. With the O2M style map, the localization performance by matching is improved than the ordinary occupancy grid map due to the additional semantic features.

C. Topological Local Object Occupancy Map

The TLO2M is a topological map representation of a path which uses the O2M as the nodes of the graph. The goal of using TLO2M is to build a sparse memory of a trajectory that only contains information, the O2M, near *representative* positions rather than memorizing the map of the entire trajectory. Therefore, it is important to make a good node selection policy to chooses *representative* positions that can express the trajectory well. Then, by what criteria can we judge whether a position is *representative* or not? Considering the homing navigation task, the node points on the original trajectory are used as subgoals of the agent navigating to the initial point. If an agent can clearly recognize that it reaches a certain location, that location can be role as a good subgoal for the navigation. From this point of view, we select positions that can be well localized as the node positions of the TLO2M. The following sections describe the details of the node selection policy.

1) Structural localizability: As mentioned above, whether a localization at a position work well or not, which we call *localizability* of a position, is an important condition of choosing node positions. In this paper, we focus on the structural uniqueness of an O2M to measure the localizability of a position. If we have a dataset of O2M observations and their matching scores of every pair, the observation with a unique structure will have low matching scores with others. With this idea, we propose a method to measure structural localizability.

$$\begin{aligned} L(o_k) = softmax(M(o_k, o_1), M(o_k, o_2), \dots, \\ M(o_k, o_k) \dots, M(o_k, o_n))[k], \quad o_1, ..o_n \in O \end{aligned} \tag{1}$$

We call $L(o_k)$ as the *localizability score* of an O2M observation o_k obtained from the RGB-D image I_k. Here, $M(o_i, o_j)$ is a matching score between o_i and o_j by using the method suggested in [5], where O is a dataset of O2M. The matching score of o_k has maximum value in self-matching, $M(o_k, o_k)$, and the value of $M(o_k, o_l)$ increases as o_k and o_l have similar structure. That is, if there are fewer structures similar to o_k in O, $L(o_k)$ increases. Therefore, if the value of $|O|$ is large enough, $L(o_k)$ can well express how unique is the given structure generally.

However, since calculating L requires a large dataset to compare, it is not realistic to use it to determine whether or not the current observation is an impressive point by measuring at inference time. To solve this problem, we train a neural network ϕ that can infer the localizabilty score with a single observation input. The ϕ is a CNN-based binary classification network that determines whether the observation is structurally unique or not. To train ϕ, we first collected a ground truth dataset consist of the pairs of I_k, o_k and $L(o_k)$. Here, we use large enough randomly sampled subsets of the O, $O(k)$, to calculate $L(o_k)$. The ϕ is trained in a supervised manner with the input of I_k and the binary target set by thresholding $L(o_k)$. As a result, the trained ϕ can infer the structural localizability of the corresponding position.

$$\hat{L}(o_k) = \phi(I_k), \quad \hat{L}(o_k) \in \{0, 1\} \tag{2}$$

2) Feature similarity: In addition to the structural uniqueness, whether the nodes are possible to distinguish each other well is also important to the localization. Here, we use unsupervised RGB-D image features to measure the similarity of two observations. We train a neural network, called *feature similarity network* (ψ) , to get and compare the similarity of RGB-D image features of the robot navigation data by using the unsupervised representation learning method called SimSiam [2]. The feature similarity network outputs the similarity (S) of two RGB-D inputs as a score between 0 to 1.

$$S(i, j) = \psi(I_i, I_j), \quad S(i, j) \in [0, 1] \tag{3}$$

We choose the nodes of the TLO2M to have low similarity score between the neighboring nodes.

3) Node selection algorithm: The proposed node selection algorithm considers both structural localizability and feature similarity. Since we use the trained ϕ and ψ networks, we can get the structural localizability and feature similarity with the online observations of the agent's initial trajectory. The proposed node selection algorithm is described as below.

Here, $dist(t_i, t_j)$ is a expected distance between two points based on the actions $a(t_i), \dots, a(t_j)$. $Integrate(o_c, o(t_k))$ represents the addition of two O2M o_c and $o(t_k)$ of grid values of the matched locations with reference to the relative expected location of t_k on o_c. Since the $\hat{L}$ is trained to learn general structural uniqueness, there is a phenomenon that the neighbors of the position which are classified as high localizability are also classified as high localizability. This is because there is no large difference of observed structure during doing a small number of actions. If all of these nearby points are used as nodes, the map will be inefficient because of the large overlaps, so we set a threshold th_{dist} to guarantee

Algorithm 1 Node selection algorithm

Result: Nodes N

Original trajectory : $\mathbf{T} = \{t_1, \ldots, t_n\}$,
RGB-D observation : $I(t_k)$, O2M : $o(t_k)$, action : $a(t_k)$ at position t_k
Current O2M : o_c
Node set : N
$o_c \leftarrow o(t_1)$ $N \leftarrow N \cup o(t_1)$
$LastNode \leftarrow t_1$
for $k \leftarrow 2$ *to* n **do**
  if $dist(t_k, LastNode) < th_{dist}$ **then**
    $o_c \leftarrow Integrate(o_c, o(t_k))$
  else
    if $\hat{L}(I(t_k)) == 1$ **or** $S(t_k, LastNode) < th_{sim}$ **then**
      $N \leftarrow N \cup o_c$, $LastNode \leftarrow t_k$
    end
  end
end

Fig. 1. An example of a TLO2M. The left figure shows the entire O2M map of the trajectory and the right one is the corresponding O2M. The colored grid represent the most major object classes in that position. The cyan dots of the left map corresponds to the node positions of TLO2M of right side. The edges represent the rough relative position of the nodes. The blue dot represent the initial point.

the minimum distance between the nodes. We also use the expected relative position obtained by the action sequences to set the distance and direction of the edges between nodes of the TLO2M. Although the relative positions are not accurate due to the actuation noise, it can give rough information of where to go to the agent navigating through the nodes. Figure 1 shows an example of the TLO2M.

D. Homing with TLO2M

The basic strategy of homing path planning using TLO2M is to do step-by-step path planning with nodes as subgoals.

TABLE I
EXPERIMENTAL RESULTS OF THE HOMING NAVIGATION IN THE GIBSON ENVIRONMENTS WITH LOW INPUT FREQUENCY

	Success	Avg path length	Avg sparsity
ORB-SLAM2 [8]	failed	failed	failed
O2M w/o node split	0.909	51	1
TLO2M w/o object class	0.939	51	0.105
TLO2M	0.955	51	0.105

The nodes of TLO2M are the O2M form which contains information about free space and obstacles around the node position, and edges contain information about rough relative position between nodes. Therefore, the location of the center position of the neighboring node can be localized in the node by using edge information. With the current O2M and the next subgoal position, we can plan a path to the subgoal from the current position by using the A* algorithm. Although the relative goal positions are not accurate, the agent can arrive near the subgoal position, which is in the range of the next node O2M. Therefore, after executing the obtained actions of the path plan, the agent localizes and updates its position on the O2M of the next node. Since the localized position of the updated agent may differ from the true position of the target subgoal node position, the agent repeatedly plans and runs the path to the true next node position until the expected distance becomes less than a threshold. The homing navigation is done by repeatedly reaching the next node position by the above process until the agent reaches the initial position of the original trajectory.

IV. EXPERIMENTS

In the experiments, we use Habitat simulator [10] based on Gibson dataset [11]. The simulation environment consists of indoor scenes of 32 different houses which split as 20 for train, and 12 for test. We use 14 object classes which are contained in the indoor environments for the feature of the O2M: chair, couch, potted plant, bed, dining table, toilet, TV, laptop, microwave, oven, sink, refrigerator, clock, and vase. For training $\hat{L}(o_k)$ network, we collected localizability score dataset from 13,566 numbers of panoramic RGB-D images captured from the Gibson dataset. We used the pretrained Mask R-CNN to build O2M for each observation, then obtained the localizability score for each observation by using equation (1). Here, the $L(o_k)$s are calculated with respect to the randomly sampled 1,024 observations. These panoramic RGB-D images are also used to train the feature similarity network. To experiment with a realistic agent, we adopt the actuation noise model suggested in [6] rather than using an accurate action model. We also assume the low-frequency panoramic RGB-D inputs that can be considered in a mobile robot with limited resources, which means the action of the agent is discrete as forwarding step size $0.25m$ and 30° of the rotation angle.

The results of the homing navigation in the Gibson environments is described in Table I. In Table I, the average sparsity is calculated as the average number of nodes divided by the average path length. The results show that the graph

representation of the TLO2M gives not only memory efficiency but also performance advances. This is because, in the TLO2M form, the localization steps only done near the nodes which are selected to be well localized. The result of the TLO2M without object class shows that the O2M helps localization than the ordinary occupancy grid map.

Also, we tried the widely used visual SLAM algorithm such as ORB-SLAM2 [8], but we found out that this algorithm fails when the input frequency is low like our settings due to the low matching features.

V. Conclusion

We propose a novel visual path memory map called the TLO2M. The TLO2M utilizes the advantages of graph map and grid map by using semantic features. The proposed method shows a 0.955 success rate of the homing navigation task at the Gibson environments. Since the proposed algorithm works well with the low input frequency, it can be used applied to mobile robots. The concept of the localizability-based map memory can be expanded to future works.

ACKNOWLEDGMENT

This work was partly supported by the Institute of Information & communications Technology Planning & Evaluation(IITP) grant funded by the Korea government(MSIT) (No. 2019-0-01309, Development of AI Technology for Guidance of a Mobile Robot to its Goal with Uncertain Maps in Indoor/Outdoor Environments).

References

[1] Devendra Singh Chaplot, Dhiraj Gandhi, Abhinav Gupta, and Ruslan Salakhutdinov. Object goal navigation using goal-oriented semantic exploration. In *Proceedings of the Advances in Neural Information Processing Systems (NeurIPS)*, 2020.

[2] Xinlei Chen and Kaiming He. Exploring simple siamese representation learning. *arXiv preprint arXiv:2011.10566*, 2020.

[3] Andrew J Davison, Ian D Reid, Nicholas D Molton, and Olivier Stasse. Monoslam: Real-time single camera slam. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 29(6):1052–1067, 2007.

[4] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. Mask r-cnn. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, 2017.

[5] Joao F Henriques and Andrea Vedaldi. Mapnet: An allocentric spatial memory for mapping environments. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018.

[6] Ashish Kumar, Saurabh Gupta, David Fouhey, Sergey Levine, and Jitendra Malik. Visual memory for robust path following. In *Proceedings of the Advances in Neural Information Processing Systems (NIPS)*, 2018.

[7] Xiangyun Meng, Nathan Ratliff, Yu Xiang, and Dieter Fox. Scaling local control to large-scale topological navigation. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, 2020.

[8] Raul Mur-Artal and Juan D Tardós. Orb-slam2: An open-source slam system for monocular, stereo, and rgb-d cameras. *IEEE Transactions on Robotics*, 33(5):1255–1262, 2017.

[9] Nikolay Savinov, Alexey Dosovitskiy, and Vladlen Koltun. Semi-parametric topological memory for navigation. In *Proceedings of the International Conference on Learning Representations (ICLR)*, 2018.

[10] Manolis Savva, Abhishek Kadian, Oleksandr Maksymets, Yili Zhao, Erik Wijmans, Bhavana Jain, Julian Straub, Jia Liu, Vladlen Koltun, Jitendra Malik, Devi Parikh, and Dhruv Batra. Habitat: A Platform for Embodied AI Research. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 2019.

[11] Fei Xia, Amir R. Zamir, Zhiyang He, Alexander Sax, Jitendra Malik, and Silvio Savarese. Gibson Env: real-world perception for embodied agents. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018.