2020 20th International Conference on Control, Automation and Systems (ICCAS 2020)
Oct. 13∼16, 2020; BEXCO, Busan, Korea

Path-Following Navigation Network Using Sparse Visual Memory

Hwiyeon Yoo[1], Nuri Kim[1], Jeongho Park[1] and Songhwai Oh[1*]

[1]Department of Electrical and Computer Engineering and ASRI, Seoul National University, Seoul 08826, Korea,
hwiyeon.yoo@rllab.snu.ac.kr, nuri.kim@rllab.snu.ac.kr, jeongho.park@rllab.snu.ac.kr, songhwai@snu.ac.kr
[*] Corresponding author

Abstract: Following a demonstration path without observing exact location of an agent is a challenging navigation problem. Especially, considering the probabilistic transition function of the agent makes the problem hard to solve with an exact action decision, so learning-based approaches have been used to solve this task. For example, a previous method by Kumar and Gupta et al., robust path following network (RPF), is a neural-network-based method using visual memories of the demonstration. Although the RPF shows good performances on the path-following task, it does not consider the efficiency of the visual memory since it requires the entire visual memory of the demonstration. In this paper, we propose a path-following network using sparse memory of the demonstration path that can deal with various sparsity of the visual memory. For each time step, the proposed network makes soft attention on the sparse memory to control the agent. We test the proposed model on the Habitat simulator using MatterPort3D dataset with various sparsity of memory. The experimental results show that the proposed method achieves 81.9% of success rate and 73.7% of SPL on a model with 0.8 memory sparsity, and also the results of the models with other memory sparsity achieve reasonable performances compare to the baseline methods.

Keywords: Visual Navigation, Deep Learning, Sparse Memory

1. INTRODUCTION

Imagine a mobile service robot that is set in a new environment. The mission of the robot is covering a service path based on a given demonstration path. If the robot is not able to access its exact location, for example in an indoor environment, it should follow the demonstration path by using only visual observations it gets. This path-following task may seem easy because one can think a naive solution: following the sequence of action of a demonstration exactly. However, actuation noises make it impossible for an agent to reproduce the exact same path even if the agent performs the same actions.

One classical approach to solve this problem is building a 3D map near demonstration path by using SLAM algorithm. While the approach localizes the agent and predicts the best action based on the reconstructed map, it is an overly resource-consuming method deriving the precise map. Rather than the precise map, knowing sequential visual and action information of the demonstrator might be more important and also enough information for the path-following. For this reason, approaches using deep neural networks are adopted to focus on the visual and action information of a path.

There have been studied a number of learning-based approaches which concentrate on solving navigation problems [3–8, 10, 12, 13]. Most of these studies aim to train their model to predict appropriate actions based on the online visual observations which have been obtained by the current agent. Among these current-agent-based navigation algorithms, Kumar and Gupta et al. [5] proposed a visual-memory-based path following algorithm, robust path following network (RPF). The RPF aims to control an agent to follow a demonstration path in a new environment by using a visual memory, which is obtained from a demonstrator, not the agent currently controlling. Although the RPF achieved good performances on the path-following task, it did not consider memory efficiency since it required the entire visual memories of the demonstration path. Memory efficiency can be a problem, for example, embedded devices like mobile service robots do not have enough memory budget to carry the entire visual memories.

To address this issue, we propose a path-following network using only sparse memory of the demonstration path. The proposed model consists of convolutional neural networks (CNN) and recurrent neural networks (RNN) which are trained by end-to-end learning. The proposed method can deal with various sparsity of visual memory without changing the model, and the experimental results show reasonable performance drop according to the sparsity of the memory.

The contributions of the paper are summarized as follows:

- We propose a learning-based path-following model using only sparse visual memory.
- The path-following performance of the proposed model tested in the simulation environments of indoor scenes is satisfactory considering the sparsity of memory.

2. RELATED WORK

There have been many studies dealing with visual navigation tasks using visual memory [3–5, 8, 12]. 'Memory' is defined differently in each paper suit for their goal. [4] proposed a mapping model and a planning model for point goal navigation. They built a spatial free space map as a memory. [8] proposed to build a topological graph map as a memory to help navigation tasks. [3] used attention by using transformer network. [11] on the previous trajectory of the current agent as a memory. [12] built a memory as a relational graph of indoor locations to use as a prior knowledge for navigating indoor an environment. [5] used observations of a demonstration path as a memory which is used to control an agent who wants to follow the demonstration path. The problem setting of the model proposed in [5], the RPF, is similar to this paper, however [5] does not consider efficiency of the memory since it uses the entire observations of the demonstration path. Different from the above methods, we propose a learning-based path-following navigation model which also considers the efficiency of memory.

3. METHOD

3.1 Problem Setup

The goal of the proposed model is to follow a demonstration path $\boldsymbol{p} = \{p_1, \ldots, p_n\}$ in an unseen environment E by using only sparse memory of the demonstration. The problem includes the information observed (observation) at each point in the path; in this paper, the observation on a point (o_t) is same as a first-person-view RGB image at the point. For each time step, the path-following network makes soft attention on the sparse memory features based on an attention point. The network then derives the next action, $\hat{a}_t$, and the next attention point by referring the attended sparse memory and a current observation, $\phi(o_t)$, where ϕ is an RGB image encoding CNN.

3.2 Network Architecture

The proposed model is a controller of the agent to follow the demonstration path. It uses the sparse memory of the demonstration as follow:

$$SP(\boldsymbol{p}) = \{M, \boldsymbol{a}, M^*\}, \tag{1}$$

where $M = \{m_1, \ldots m_l\}$ is a sparse visual memory set among n-length path, M^* is a list of indices of observations used in the memory, and $\boldsymbol{a} = \{a_1, \ldots, a_n\}$ represents the actions of the demonstrator. Note that we use all demonstration actions including actions without corresponding visual memories to leverage information of the relative location of each visual memory.

When the agent follows the path, an attention mechanism is used over $SP(\boldsymbol{p})$. At each time step t, an attention pointer η_t is used to get softly attended M and hardly attended $\boldsymbol{a}$. The attended sparse memory μ_t is following as,

$$\mu_t = \psi(\sum_j m_j e^{-|\eta_t - M^*(j)|}, \boldsymbol{a}_{\eta_t}),\ j = 1, \ldots, l \tag{2}$$

where $\boldsymbol{a}_{\eta_t} = \{a_{\lfloor \eta_t \rfloor - k}, \ldots, a_{\lfloor \eta_t \rfloor + k}\}$ is a subset of $\boldsymbol{a}$ based on the hyperparameter k, and ψ is a trainable fully-connected layer. The path following network π is realized as a gated recurrent unit (GRU [2]) as follow:

$$h_{t+1}, b, \hat{a}_t = \pi(h_t, \mu_t, \phi(o_t), \boldsymbol{a}_{\eta_t}) \tag{3}$$

$$\eta_{t+1} = \eta_t + tanh(b) \tag{4}$$

where h_t is an internal state of the GRU, o_t is the current observation, and b is the increment of the attention pointer η. The initial settings are $h_1 = \mathbf{0}$ and $\eta_1 = 1$. The overview path following network is shown in the Figure 3.1

3.3 Training

We apply an imitation learning to train the proposed network. For collecting expert data, we sampled perturbed paths from demonstration paths. The imitation learning loss L_{il} is a cross-entropy between the expert's action and the path-following agent's action:

$$L_{il} = \sum_t a_t^{\text{ex}} log \hat{a}_t, \tag{5}$$

where a_t^{ex} is the action of expert in step t.

4. EXPERIMENTS

4.1 Experimental Settings

In the experiments, we use Habitat simulator [9] based on MatterPort3D dataset [1]. The simulation environment consists of indoor scenes of 90 different houses which split as 61 for train, 11 for validation and 18 for test. For demonstration paths, we collect optimal paths from the point goal navigation dataset provided by [9] which consists of initial point and goal point pairs and the optimal paths between two points. Action space of the agent is as follows: move forward for $0.3m$, turn left or right for 20°. We also collect perturbed paths with actuation noises, which follow the same initial point and goal point with the corresponding demonstration path. These data become the path-following data and be used to train the proposed network by imitation learning. We sample an actuation noise from $\mathcal{N}(0,\ 0.5^2)$ (m) for forward movement and $\mathcal{N}(0,\ 1^2)$ $(radian)$ for rotation.

We crop the demonstration data by length of 30 steps (n), and set a maximum length of the path-following data by 50 steps. The size of the sparse visual memory, l, is determined by multiplying n with a sparsity defined in 0 and 1. We choose a sparsity of the model before training, and fix it during the training and testing time. The sparse memories are sampled with even intervals to fit the goal sparsity, where $M^* = \{\lfloor i \times \frac{n}{l-1} \rfloor | i = 1, \ldots, l-1\} \cup \{1\}$.

In the experiments, we use success rate and success weighted by normalized inverse path length (SPL) as evaluation metrics. The success of an agent is defined as reaching within 2 steps or 10% of the initial distance to goal, whichever is larger.

4.2 Experimental Results

The experimental results of the path-following task in the test environments is represented in Table 1. We test the proposed network with sparse memory of sparsity 0.4, 0.6, and 0.8. For baseline models, we take RPF [5] which uses entire visual memories for path-following and the action-only-version of RPF which does not use visual memory at all. Each model in the table is trained and tested independently on the proposed dataset.

Table 1 shows that the performances of our models are between the two baseline models which means that the proposed model exploits the sparse visual memory effectively. Note that the performance of RPF is the upper bound of the proposed model since RPF uses entire visual memories. In addition, the results show that performance

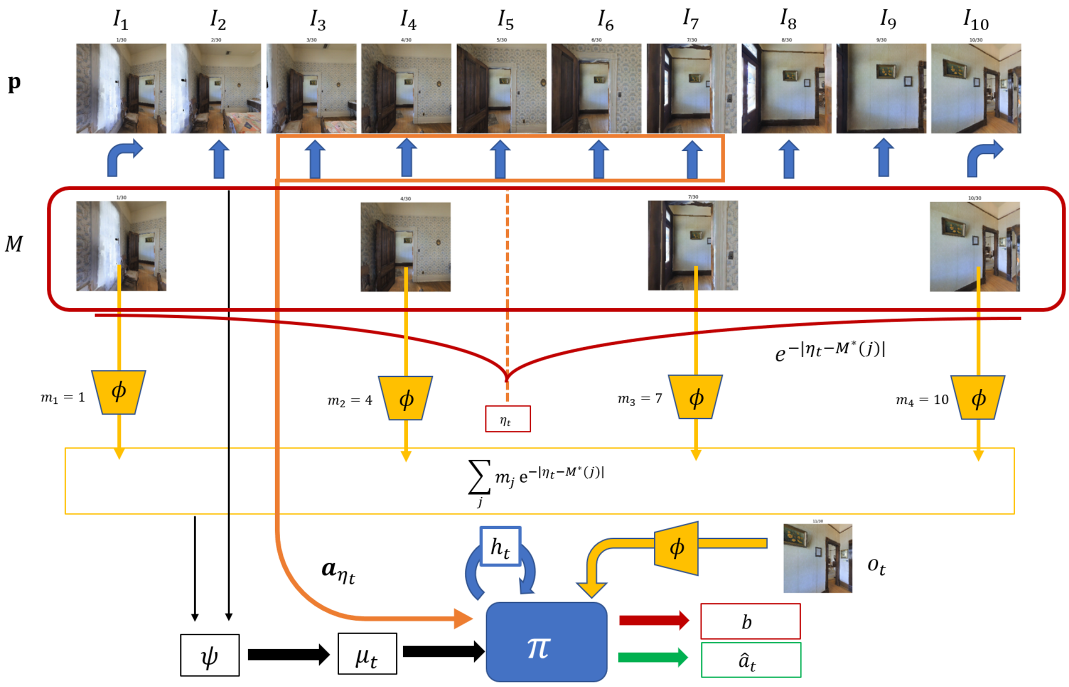

Fig. 1 Overview of the proposed model. The sparse memory,M, is evenly sampled from the path $\boldsymbol{p}$. Based on the attention pointer η_t which increase gradually during training, soft attended sparse visual memories and the hard attended action list are fused by ψ. The fused feature μ_t, the hard attended action lists $\boldsymbol{a}_{\eta_t}$, and the current observation $\phi(o_t)$ are input of the policy GRU π. π outputs the next action $\hat{a}_t$ and attention increment b.

Table 1 Performance of the path-following task with various sparsity of visual memory which are tested on the test environment of the Habitat simulator rendering the MP3D dataset

	Sparsity 0.4		Sparsity 0.6		Sparsity 0.8		RPF only action [5]		RPF [5]	
	Success	SPL	Success	SPL	Success	SPL	Success	SPL	Success	SPL
SPF	0.782	0.683	0.803	0.724	0.819	0.737	0.782	0.648	0.833	0.752

increases when the memory size increase, which is in line with the common intuition.

5. CONCLUSIONS

In this paper, we have presented a learning-based path following navigation model using a sparse memory of a demonstration. The proposed sparse-memory-based model achieves memory efficiency with less drop of path-following performance. The proposed model can be used to agents with limited memory budget, such as mobile robots.

ACKNOWLEDGMENT

This work was supported by 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] Angel Chang, Angela Dai, Thomas Funkhouser, Maciej Halber, Matthias Niessner, Manolis Savva, Shuran Song, Andy Zeng, and Yinda Zhang. Matterport3d: Learning from rgb-d data in indoor environments. *International Conference on 3D Vision (3DV)*, 2017.

[2] Kyunghyun Cho, Bart Van Merriënboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. Learning phrase representations using rnn encoder-decoder for statistical machine translation. 2014.

[3] Kuan Fang, Alexander Toshev, Li Fei-Fei, and Silvio Savarese. Scene memory transformer for embodied agents in long-horizon tasks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 538–547, 2019.

[4] Saurabh Gupta, James Davidson, Sergey Levine, Rahul Sukthankar, and Jitendra Malik. Cognitive mapping and planning for visual navigation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2616–2625, 2017.

[5] 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)*, pages 765–774, 2018.

[6] Piotr Mirowski, Matt Grimes, Mateusz Malinowski, Karl Moritz Hermann, Keith Anderson, Denis

Teplyashin, Karen Simonyan, Andrew Zisserman, Raia Hadsell, et al. Learning to navigate in cities without a map. In *Proceedings of the Advances in Neural Information Processing Systems (NIPS)*, pages 2419–2430, 2018.

[7] Piotr Mirowski, Razvan Pascanu, Fabio Viola, Hubert Soyer, Andrew J Ballard, Andrea Banino, Misha Denil, Ross Goroshin, Laurent Sifre, Koray Kavukcuoglu, et al. Learning to navigate in complex environments. In *Proceedings of the International Conference on Learning Representations (ICLR)*, 2017.

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

[9] 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.

[10] William B Shen, Danfei Xu, Yuke Zhu, Leonidas J Guibas, Li Fei-Fei, and Silvio Savarese. Situational fusion of visual representation for visual navigation. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, pages 2881–2890, 2019.

[11] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Proceedings of the Advances in Neural Information Processing Systems (NIPS)*, pages 5998–6008, 2017.

[12] Yi Wu, Yuxin Wu, Aviv Tamar, Stuart Russell, Georgia Gkioxari, and Yuandong Tian. Bayesian relational memory for semantic visual navigation. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, pages 2769–2779, 2019.

[13] Yuke Zhu, Roozbeh Mottaghi, Eric Kolve, Joseph J Lim, Abhinav Gupta, Li Fei-Fei, and Ali Farhadi. Target-driven visual navigation in indoor scenes using deep reinforcement learning. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, pages 3357–3364, 2017.