Learning to Explore, Navigate and Interact for Visual Room Rearrangement

Ue-Hwan Kim*, Young-Ho Kim*, Jin-Man Park, Hwan-Soo Choi, Jong-Hwan Kim
School of Electrical Engineering, KAIST
Daejeon, Republic of Korea
{uhkim, yhkim, jmpark, hschoi, johkim}@rit.kaist.ac.kr

Abstract

Intelligent agents for visual room rearrangement aim to reach a goal room configuration from a cluttered room configuration via a sequence of interactions. For successful visual room rearrangement, the agents need to learn to explore, navigate and interact within the surrounding environments. Contemporary methods for visual room rearrangement display unsatisfactory performance even with state-of-the-art techniques for embodied AI. One of the causes for the low performance arises from the expensive cost of learning in an end-to-end manner. To overcome the limitation, we design a three-phased modular architecture (TMA) for visual room rearrangement. TMA performs visual room rearrangement in three phases: the exploration phase, the inspection phase, and the rearrangement phase. The proposed TMA maximizes the performance by placing the learning modules along with hand-crafted feature engineering modules—retaining the advantage of learning while reducing the cost of learning.

1. Introduction

Embodied AI studies learning of embodied physical interactions with surrounding environments such as visual navigation and embodied question answering [4]. Recently, tasks involving direct physical interaction with objects are drawing increasing attention [1]. Tasks that involve interaction with objects in environments include the visual room rearrangement task [9]. Intelligent agents for visual room rearrangement perform a series of interactions with objects in a cluttered environment to attain a goal state of the environment. Successful visual room rearrangement requires the agents to efficiently explore and navigate through the environment, and effectively recognize and manipulate the objects in the environment.

Formulating visual room arrangement as an end-to-end learning problem is straightforward, but such an approach likely suffers from the expensive cost of pure learning [3]. To cope with the high cost of pure learning, the previous work has relied on the imitation learning [6] only to demand millions of experience and to display unsatisfactory performance [9].

To overcome the limitations mentioned above, we propose a three-phased modular architecture (TMA) for the visual room rearrangement task. The proposed TMA locates the learning modules along with hand-crafted feature processing modules. This modular design retains the advantage of learning, reduces the cost of learning, and maximizes the performance since learning modules focus on high-level tasks and the hand-crafted feature processing modules support the learning modules with expert knowledge.

TMA conducts visual room rearrangement in three phases: the exploration phase, the inspection phase, and the rearrangement phase. In the exploration phase, the embodied agent explores and navigates through an unseen environment in its goal configuration. As a result, the agent obtains a semantic map of the environment. In the inspection phase, the agent once again navigates through the environment in its cluttered configuration given the constructed semantic map. Then, the agent obtains another semantic map of the environment. In the rearrangement phase, the agent compares the two semantic maps, plans the rearrangement process, and performs the rearrangement task.

2. Method

2.1. Exploration Phase

In the exploration phase, the embodied agent walks through an unseen environment in its goal configuration and obtains a semantic metric map.

Semantic Metric Map. We define the semantic metric map as a $K \times M \times M$ matrix where $M \times M$ denotes the map size and $K = 4+C$ represents an obstacle map, the explored area, the current agent location, the past agent locations, and C categories of semantics. We initialize the map with all zeros at the beginning of the phase and the agent stores the collected information regarding the environment

*equal contribution

in the map throughout the phase. We make an assumption that the agent starts exploration at the center of the map.

Exploration and Navigation. The agent iterates the process of predicting one long-term goal and a sequence of short-term goals for reaching the long-term goal multiple times to explore and navigate through the environment. For this, the agent utilizes two modules: a long-term goal policy module (LGPM) and a short-term goal planner module (SGPM). The agent first set a long-term goal with LGPM. LGPM receives the current semantic metric map and predicts a long-term goal using a convolutional neural network. LGPM is a learning module, and we train it using reinforcement learning [7] with reward proportional to the increase in coverage of the semantic map and decrease in the total time spent.

Next, SGPM provides a sequence of short-term goals with corresponding actions for reaching the long-term goal. SGPM uses the fast marching method [8] to compute the shortest path from the current location to the long-term goal and heuristics to determine the actions to realize the shortest path. This planning module does not require any training process so that it dramatically reduces the total number of parameters to learn.

2.2. Inspection Phase

In the inspection phase, the agent once again navigates through the environment in a cluttered configuration given the constructed semantic map from the exploration phase. After the navigation, the agent obtains another semantic metric map of the environment in its cluttered configuration. The agent uses the pair of two resulting semantic metric maps in the rearrangement phase.

For navigation in the inspection phase, the agent utilizes LGPM and SGPM as well. However, we learn another set of LGPM parameters that suit the purpose of the inspection phase. Since the purpose of the inspection phase is rapidly obtaining a semantic map of the known environment in a different state, the agent would exploit the collected knowledge in the prior phase rather than trying to explore. Thus, we put more weight on the time saving when calculating the reward.

2.3. Rearrangement Phase

Change Detection. In the rearrangement phase, the agent first detects changes between the goal configuration and the cluttered configuration by comparing the two semantic maps. The changes include the location and state of objects. A simple comparison of the pair of semantic maps would not result in successful change detection due to imperfect object detection. To deal with imperfect object detection, we identify the same objects in two semantic maps using the following metric:

$$d = w_{\text{class}} \cdot s_{\text{class}} + w_{\text{class}} \cdot s_{\text{size}}, \tag{1}$$

where s_{class} and s_{size} denote the class label similarity and the size similarity, w_{class} and w_{size} are the weights for the similarities and $w_{\text{class}} + w_{\text{size}} = 1$. We compute the class similarity using the distance between the corresponding word vectors [2] and the size similarity using L_2-norm.

Planning. After detecting the changes, the agent plans the rearrangement process. The planning process determines the order of rearranging each object which is optimal in respect of time complexity. For N matched objects from change detection, the agent evaluates $N!$ permutations of orders and selects the optimal order. Since the maximum number of different objects are 5 in our problem setting [9], this exhaustive search guarantees affordable computational and time complexities.

Rearrangement. Finally, the agent performs the rearrangement task based on the resulting plan: the agent rearranges the objects in the given order (plan). For rearranging each object, the agent utilizes the A^* path planner [10] to plan the path and a sequence of actions.

Table 1. Experiment results on the three data splits.

Split	Success	%Fixed	%E	%Misplaced
Train	0.5	1.0	1.01	1.00
Val.	0.0	0.9	1.00	1.00
Test	0.1	0.6	1.01	1.01

3. Evaluation

3.1. Settings

We use the visual room rearrangement dataset and the evaluation protocol provided by the 2021 AI2-THOR Rearrangement Challenge hosted at the CVPR 2021 Embodied AI Workshop [9]. The dataset consists of 6,000 unique rearrangement scenarios (4,000/1,000/1,000 for train, validation and test, respectively) in 120 rooms of AI2-THOR [5]. Moreover, the performance metrics include Success, %Fixed, %Energy Remaining and %Misplaced.

3.2. Results

Table 1 displays the experiment results on the three data splits. The proposed TMA shows moderate performance in visual room rearrangement. However, the performance gap between the current model and the performance for practical home service agents is still substantial.

4. Conclusion

In this work, we proposed a three-phased modular architecture (TMA) for visual room rearrangement. The proposed TMA alleviates the high cost of end-to-end learning approaches with a modular design. TMA retains the advantage of learning and reduces the cost of learning by placing the learning modules along with hand-crafted feature processing modules. TMA performs the visual room rearrangement task in three phases: the exploration phase, the inspection phase, and the rearrangement phase. We expect the proposed TMA would let the embodied AI research take one step forward towards intelligent agents realizing practical services for humans by interacting with the surrounding in an effective manner.

References

[1] Dhruv Batra, Angel X Chang, Sonia Chernova, Andrew J Davison, Jia Deng, Vladlen Koltun, Sergey Levine, Jitendra Malik, Igor Mordatch, Roozbeh Mottaghi, et al. Rearrangement: A challenge for embodied ai. *arXiv preprint arXiv:2011.01975*, 2020. 1

[2] Piotr Bojanowski, Edouard Grave, Armand Joulin, and Tomas Mikolov. Enriching word vectors with subword information. *Transactions of the Association for Computational Linguistics*, 5:135–146, 2017. 2

[3] Devendra Singh Chaplot, Dhiraj Gandhi, Saurabh Gupta, Abhinav Gupta, and Ruslan Salakhutdinov. Learning to explore using active neural slam. In *International Conference on Learning Representations (ICLR)*, 2019. 1

[4] Duan Jiafei, Samson Yu, Tan Hui Li, Hongyuan Zhu, and Cheston Tan. A survey of embodied ai: From simulator to research tasks. *arXiv preprint arXiv:2103.04918*, 2021. 1

[5] Eric Kolve, Roozbeh Mottaghi, Winson Han, Eli VanderBilt, Luca Weihs, Alvaro Herrasti, Daniel Gordon, Yuke Zhu, Abhinav Gupta, and Ali Farhadi. Ai2-thor: An interactive 3d environment for visual ai. *arXiv preprint arXiv:1712.05474*, 2017. 2

[6] Stéphane Ross, Geoffrey Gordon, and Drew Bagnell. A reduction of imitation learning and structured prediction to no-regret online learning. In *Proceedings of the fourteenth international conference on artificial intelligence and statistics*, pages 627–635. JMLR Workshop and Conference Proceedings, 2011. 1

[7] John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347*, 2017. 2

[8] James A Sethian. A fast marching level set method for monotonically advancing fronts. *Proceedings of the National Academy of Sciences*, 93(4):1591–1595, 1996. 2

[9] Luca Weihs, Matt Deitke, Aniruddha Kembhavi, and Roozbeh Mottaghi. Visual room rearrangement. *arXiv preprint arXiv:2103.16544*, 2021. 1, 2

[10] Wei Zeng and Richard L Church. Finding shortest paths on real road networks: the case for a. *International journal of geographical information science*, 23(4):531–543, 2009. 2