Pose-guided End-to-end Visual Navigation

Cuiyun Fang¹,²,a, Chaofan Zhang²,b*, Fulin Tang³,c, Fan Wang²,d, Yihong Wu³,e, and Yong Liu²,f

¹Institute of Physical Science and Information Technology, Anhui University, Anhui 230000, China
²Anhui Institute of Optics and Fine Mechanics, Hefei Institutes of Physical Science, Chinese Academy of Sciences, Anhui, 230000, China
³The National Laboratory of Pattern Recognition, Institute of Automation, Chinese Academy of Sciences, Beijing, 101408, China
a-e-mail: fangcy@mail.ustc.edu.cn, c-e-mail: fulin.tang@nlpr.ia.ac.cn, d-e-mail: wangf@aiofm.ac.cn, e-e-mail: yhwu@nlpr.ia.ac.cn, f-e-mail: liuyong@aiofm.ac.cn

*Corresponding author: b-e-mail: zcfan@aiofm.ac.cn

Abstract. End-to-end visual navigation based on deep reinforcement learning (DRL) has recently attracted much attention. For most existing navigation methods, a robot moves along only fixed directions (e.g., up, down, left and right) on a grid. Obviously, they are not flexible and efficient, which worsens the navigation performance (i.e., the distance of movement and times of rotation). To address this problem, we propose a novel pose-guided end-to-end visual navigation framework, which is flexible and efficient. In the pose-guided navigation framework, a robot can move along arbitrary directions, which are determined by poses between adjacent objects. Further, to select a proper motion and finally form an optimal path, we propose a DRL based action-selected strategy, where a dynamic action select space on the basis of deep siamese actor-critic network is developed. Besides, to validate the proposed method, we propose a novel pose-guided dataset. Experimental results demonstrate that the proposed method outperforms the state of the arts in both flexibility and efficiency.

1. Introduction

Visual navigation plays an important role in diverse fields such as mobile robots[1], unmanned aerial vehicles(UAV)[2] and unmanned ground vehicles(UGV)[3]. Generally, traditional navigation approaches have modular pipelines [4]: firstly building maps [5], then planning paths[6] and finally controlling movements[7]. These map-based navigation approaches rely heavily on the qualities of the maps and are time-consuming. Compared to the map-based navigation approaches, end-to-end visual navigation based on DRL can directly obtain control policies according to input images, which are concise and efficient [8]–[10]. Thus, end-to-end visual navigation based on DRL has attracted much interest in recent years.

Although end-to-end navigation has achieved great success in unknown environment navigation [11]–[13] and dynamic avoiding collision[14], most existing methods interact with the environment by fixed motions (e.g., up, down, left and right) and are restricted to move on a grid [15]–[18], which degrades navigation flexibility and efficiency. Because of these problems, end-to-end navigation has not
been utilized widely, which motivates us to develop a flexible and efficient end-to-end navigation method.

In this paper, we propose a novel pose-guided end-to-end visual navigation framework, where a robot can move along arbitrary directions according to poses between adjacent objects. Intuitively, the proposed method is more flexible and anthropomorphic, as shown in Fig. 1. Moreover, the DRL based action-selected strategy is leveraged to learn the policy of selecting optimal actions, which is improved based on deep siamese actor-critic network[19] by setting dynamic action select space. In order to accomplish the pose-guided navigation framework, we make a pose-guided dataset. As far as we know, the pose-guided dataset is proposed firstly. To fill the aforementioned void in the community, we public our proposed dataset at https://github.com/CASHIPS-ComputerVision/Pose-guided-dataset.

The novel contributions are summarized as follows.
1) A novel flexible pose-guided navigation framework is proposed, where a robot can move along arbitrary directions determined by poses between adjacent objects.
2) A DRL based action-selected strategy is developed to select a proper motion and finally form an optimal path, where a dynamic action select space is set by using deep siamese actor-critic network.
3) To validate the proposed method, a novel pose-guided dataset is proposed. Experimental results demonstrate that the proposed method outperforms the state of the arts in both flexibility and efficiency.

The rest of this paper is organized as follows. The related work is introduced in Section 2. The detail of the proposed pose-guided end-to-end visual navigation method is presented in Section 3. The experimental results are provided in Section 4. And in Section 5, we describe conclusions.

Fig.1. Navigation trajectories of different methods. Left: the trajectory(red) of end-to-end visual navigation [19] and right: the trajectory(blue) of the proposed method.

2. Related Work
In this section, we describe the related work on visual navigation. There are two groups: traditional visual navigation and end-to-end visual navigation.

2.1 Traditional Visual Navigation
Simultaneous Localization and Mapping (SLAM) is the key technology for mapping. Noticeable works have been yielded, such as PTAM [20], ORB-SLAM [5], [21], [22], LSD-SLAM [23] and MSCKF [24]. Based on a map, path planning can be employed. Various path planning methodologies have been provided, which can be divided into classical and reactive method. Cell decomposition (CD) [25], roadmap approach (RA) and artificial potential field (APF) are the classical methods. Reactive methods include Genetic algorithm (GA), fuzzy logic (FL), particle swarm optimization (PSO) and ant colony optimization (ACO)[26].

2.2 End-to-end Visual Navigation
End-to-end visual navigation bypasses the modular steps and directly learns navigation strategies with deep networks. Pomerleau et al. [29] proposed the pioneer ALVINN system in 1989, which uses multilayer perception to learn the directions a vehicle should steer. Afterwards, deep learning based end-to-end navigation have achieved excellent success, due to the impressive convolutional neural networks (CNNs) and powerful computation power. Brahmbhatt et al. [30] presented a city-level end-to-end navigation system based on CNN, which uses local visual streetscape images for navigation. Xu et al.
demonstrated an end-to-end method, which obtains future vehicle egomotion from a large-scale video dataset based on CNN.

Due to the significant recent development in DRL, DRL based end-to-end navigation methods were presented[9],[10]. Among these works, A3C[32] and PPO[33] are two widely utilized DRL algorithms, because A3C has fast computing speed for parallel tasks and PPO has excellent performance for continuous control tasks. Mirowski et al.[34] presented an end-to-end navigation system by training an Asynchronous Advantage Actor-Critic(A3C) agent in a 3D maze. Luo et al.[35] developed a vision-based target-driven expert policy by training PPO algorithm in the Habitat simulator. In addition to DRL algorithms, datasets play an important role in end-to-end navigation. Several datasets were proposed to improve navigation performance, for example AI2-THOR[36], Streetlearn[37] and AdobeIndoorNav[38]. With the assistant of AI2-THOR[36], Zhu et al.[19] proposed a deep siamese actor-critic model, which improves the generalization of end-to-end navigation.

In summary, DRL based end-to-end methods have gained much attention so far. Different from most existing methods move on a grid, we propose a novel flexible pose-guided end-to-end visual navigation method, by using the poses between adjacent objects to guide movement arbitrarily.

3. Method
In this section, we detail the proposed pose-guided end-to-end visual navigation framework, as shown in Fig. 2. First, the observation and target images are the input. Then, according to poses between adjacent objects from the proposed pose-guided dataset, the action space is generated. Afterwards, we utilize the DRL based action-selected strategy to select the optimal actions. Finally, all selected actions form an optimal navigation path as the output.

In the proposed framework, visual navigation is carried out according to poses between adjacent objects. A pose contains the distance and the angle between an object and a robot. Under the guidance of pose information, we can determine the interactive actions of the robot. For example, if the pose is \((-\alpha, d)\), we set the robot's interactive action to turn left \(\alpha\) degrees and then walk forward \(d\) meters, as shown in Fig. 3. To complete the pose-guided navigation framework, we make a pose-guided dataset because there is no dataset suitable for the framework.

![Fig. 2. The pipeline of the proposed pose-guided navigation framework.](image)

![Fig. 3. The pipeline of the proposed pose-guided navigation framework.](image)
3.1. Pose-guided Dataset

The proposed dataset is the discrete representation of the environment, which selects the reachable objects as discrete points. At each discrete point, RGB and depth images of the surrounding scenes are captured. All RGB images represent the overall state space of the robot. And all depth images are input into VoteNet[39] to obtain poses between adjacent objects. VoteNet is the state-of-the-art network for 3D target detection, which uses the deep Hough voting model to achieve 3D object detection in a point cloud. The poses are contained by the output 3D bounding box of VoteNet. The process of building the pose-guided dataset is shown in Fig. 4.

![Fig.4. Process of building pose-based dataset.](image)

Based on the proposed pose-guided dataset, we let the robot move ceaselessly in the environment from a random starting point to the selected target point. Then, rewards of different actions are obtained and action-selected policies are generated. In this process, the proposed framework translates poses into motions directly by visiting observation images and getting the corresponding poses from the proposed dataset. Thus, the pose-guided motions are set as the interactive actions of the robot. The robot executes actions to obtain the rewards during the training process and moves along arbitrary directions.

3.2. Action-selected Strategy

In this section, we propose a DRL based action-selected strategy to learn the policy that enables the robot to select the optimal action at each state and generate the navigation path. The strategy is implemented in the proposed pose-guided dataset.

Generally, in the DRL based action-selected methods, the robot learns the optimal action policy by interacting with the environment. The optimal action policy is learned by maximizing the expected sum of rewards:

$$V_\pi = \mathbb{E}_\pi [\sum_{t=0}^{\infty} \gamma^t r_t]$$

Where $\gamma \in [0,1)$ is the discount factor, $r_t$ is the reward obtained by the robot executing $a_t$ in the state of $S_t$, $S_t$ is the state at time $t$ and $a_t \sim \pi (\cdot | S_t)$ is the action generated by following some policies $\pi$.

Inspired by the target-driven navigation model[19], we propose a novel DRL based action-selected strategy. The original target-driven navigation model combines siamese network and actor-critic network to realize end-to-end visual navigation. In order to improve it further, we set dynamic action space and improve the reward function. These improvements include three parts: action space, reward function and network architecture. We describe them in detail as follows.

**Action space.** By setting a dynamic action space, the robot can select actions flexibly at each state. To constrain different possible states and allow off-line batch training, we consider the actions as discrete movements in the exploration space. Different from actions in other navigation methods, actions in the proposed method are guided by poses between adjacent objects. Moreover, we set two basic actions at each state: turn left $\pi/2$ and turn right $\pi/2$. When the robot executes the action (i.e., rotating $\alpha$ degrees and walking $d$ meters) to get into next state, we rotate the robot in reverse to ensure that the
observation of the robot belongs to the proposed dataset. In order to make reward dimensions consistent, we supplement the action space to the maximum action space capacity in the scene and replace the missing actions by a fixed value.

**Reward function.** To accomplish the low navigation cost, we adopt a straight-forward reward function design. When the robot reaches the target state, it will obtain a reward (10.0). And we added a small amount of time penalty (-0.01) as an instant reward. To prevent the robot from choosing an invalid action, we set the reward for implementing the action to -100. Furthermore, we add a step to identify whether the action is valid in navigation.

**Network architecture.** The architecture of the proposed DRL based action-selected network is shown in Fig. 5. In the network, we take observation and target images as the inputs. Two ResNet-50 networks and a full connection layer consist of the generic siamese network. The ResNet-50 and full connection layers share parameters. Finally, a dynamic policy and a reward are output through two FC layers.

![Fig.5. The network architecture of the proposed action-selected strategy.](image)

### 4. Experiment

In this section, we compare our method with the state-of-the-art methods: A3C [32], PPO [33] and a target-driven navigation method [19]. Since existing public datasets do not contain poses between adjacent objects, we perform experiments in the proposed pose-guided dataset, as shown in Fig. 4. There are four scenes in the proposed pose-guided dataset, including a living room, a laboratory, an apartment and a break room. For [32], [33] and [19], vision navigation is implemented on a grid. Thus, in the same scenes with the proposed pose-guided dataset, we construct a grid and train them on the grid.

We train the proposed method on a Nvidia GeForce GTX 1050 GPU. We describe the training as follows. For the proposed network, we utilize the adam optimizer with learning rate 0.001 and the network is trained for 10000 epochs. An episode only terminates when the step arrives 10000. We train the robot for 10000000 steps totally. During the training process, we select five destinations as the training set for each image set. In the test, we randomly select 100 starting states to reach a given destination. Navigation paths are the output.

Firstly, we evaluate the efficiency and flexibility. In terms of efficiency, we validate the efficiency quantitatively by comparing the navigation cost, including rotation times and movement distance. In terms of flexibility, the flexibility is analysed qualitatively by comparing the navigation paths. Finally, in order to demonstrate the comprehensive navigation performance, we perform experiments to compare the Success Rate (SR) and the Success weighted by Path Length (SPL).

#### 4.1. Efficiency

We train and test the proposed method, [32], [33] and [19] in the proposed dataset. We train and test the proposed method, [32], [33] and [19] in the proposed dataset. And the average navigation costs of 100 episodes, including movement distance and rotation times, are compared. As we can see from Fig. 6, the navigation costs of our method are smaller than other three methods in terms of rotation times and moving distance. Specifically, in the apartment, the rotation time of our method is only 0.5 times and the movement distance is less than 1/3 times than that of other methods, respectively. In the laboratory, the scene is more complex than other three scenes. Although our method has more rotation times, its
movement distance is far less than that of other methods. In summary, the proposed pose-guided end-to-end navigation method has lower navigation cost. The reason is that the movements in our method are determined by the poses between adjacent objects other than on a grid.

4.2. Flexibility
We demonstrate the flexibility qualitatively by comparing trajectories of different methods. Their trajectories are shown in Fig. 7. The visualize result shows the moving trajectories of the robot directly for different methods in the same starting and end point. Different from other methods whose trajectories are constrained on the grid, the trajectories of our method are guided by poses. Therefore, the robot can have flexible action choices at each state.

4.3. Navigation Performance
In order to further validate the comprehensive navigation performance, we evaluate the Success Rates (SR) and Success weighted by Path Length (SPL) [40] of different navigation methods. SPL measures the robot navigation performance as
\[ \frac{1}{N} \sum_{i=0}^{N} s_i \frac{l_i}{\max(p_i, l_i)} \]  

(2)

where \( N \) denotes the number of testing episodes, \( s_i \) is a binary indicator of success in the testing episode \( i \), \( p_i \) represents path length and \( l_i \) is the shortest path distance from to the goal.

Table 1. Success Rates (SR) and Success weighted by Path Length (SPL) of different end-to-end navigation methods.

| Method                | Scenes      | SR | SPL | SR  | SPL | SR  | SPL | SR  | SPL |
|-----------------------|-------------|----|-----|-----|-----|-----|-----|-----|-----|
| A3C [32]              | Apartment   | 100| 0.093 | 100 | 0.139 | 100 | 0.076 | 100 | 0.308 |
| Target-driven [19]    | Break room  | 100| 0.060 | 99  | 0.104 | 84  | 0.040 | 100 | 0.168 |
| PPO [33]              | Laboratory  | 100| 0.055 | 96  | 0.074 | 84  | 0.036 | 100 | 0.123 |
| Ours                  | Living room | 100| 0.058 | 95  | 0.065 | 90  | 0.042 | 100 | 0.177 |

Table I summarizes the results for the four methods in the four testing scenes. The proposed navigation strategy performs better than other methods in SR and SPL. Our method can achieve 100% success rate. The SPL of our method is above 0.1. In the apartment, the scene is simple. The SPL of our method can exceed 0.3, which is better than that of other methods. In the laboratory, the scene is complex. While SPL is just near 0.1, it is still nearly twice times that of other methods. The reason is that the complexity of the environment affects the navigation performance.

In summary, compared to the state-of-the-art methods, the proposed pose-guided navigation method is more flexible and efficient. Moreover, our method has better navigation performance in both SR and SPL.

5. Conclusion

We present a novel pose-guided end-to-end navigation method. The main advantage is that the robot can move along arbitrary directions determined by poses between adjacent objects. What's more, we make a novel pose-guided dataset to validate our method. And DRL based action-selected strategy is proposed to select actions. The results demonstrate that the proposed method outperforms state-of-the-art methods in flexibility and efficiency.

Acknowledgments

This work was supported by the Science and Technology Major Program of Anhui Province of China under Grants 202003a05020020 and 201903a05020041, Joint fund of Science & Technology Department of Liaoning Province and State Key Laboratory of Robotics, China under Grant 2020- KF22-16, Special Foundation of President of the Hefei Institutes of Physical Science under Grant YZJJ2020QN36, Open Projects Program of National Laboratory of Pattern Recognition under Grant 202100040.

References

[1] Pfeiffer M, Schaeuble M, Nieto J, (2017) From perception to decision: A data-driven approach to end-to-end motion planning for autonomous ground robots. In: IEEE international conference on robotics and automation (icra). Marina Bay Sands. pp. 114-118.

[2] Wang C, Wang J, Shen Y. (2019) Autonomous navigation of UAVs in large-scale complex environments: A deep reinforcement learning approach. IEEE Transactions on Vehicular Technology vol.68, no. 3, pp. 2124-2136, 2019.

[3] Wang H, Yuan S, Guo M, et al. (2021) Tactical driving decisions of unmanned ground vehicles in complex highway environments: A deep reinforcement learning approach. Proceedings of the Institution of Mechanical Engineers, Part D: Journal of Automobile Engineering. 235(4): 1113-1127.
[4] Yuan W, Li Z, Su C Y. (2016) RGB-D sensor-based visual SLAM for localization and navigation of indoor mobile robot. In: International Conference on Advanced Robotics and Mechatronics (ICARM), Macau. pp. 82-87.

[5] Campos C, Elvira R, Rodríguez J J G, (2020) ORB-SLAM3: An accurate open-source library for visual, visual-inertial and multi-map SLAM. https://arxiv.org/pdf/2007.11898.

[6] Maurovic I, Seder M, Lenac K. (2017) Path planning for active SLAM based on the D* algorithm with negative edge weights. IEEE Transactions on Systems, Man, and Cybernetics: Systems, pp. 1321-1331.

[7] W. Schwarting, J. Alonso-Mora, L. Paull, S. Karaman, and D. Rus. (2017) Safe nonlinear trajectory generation for parallel autonomy with a dynamic vehicle model. IEEE Transactions on Intelligent Transportation Systems, pp. 2994-3008.

[8] Shi H, Shi L, Xu M. (2019) End-to-end navigation strategy with deep reinforcement learning for mobile robots. IEEE Transactions on Industrial Informatics, pp. 2393-2402.

[9] Zhang J, Springenberg J T, Boedecker J. (2017) Deep reinforcement learning with successor features for navigation across similar environments. In: 2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Vancouver. pp. 2371-2378.

[10] Tai L, Paolo G, Liu M. (2017) Virtual-to-real deep reinforcement learning: Continuous control of mobile robots for mapless navigation. In: 2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Vancouver. pp. 31-36.

[11] Wang C, Wang J, Zhang X. (2017) Autonomous navigation of UAV in large-scale unknown complex environment with deep reinforcement learning. In: 2017 IEEE Global Conference on Signal and Information Processing (GlobalSIP), Montreal. pp. 858-862.

[12] Li H, Zhang Q, Zhao D. (2019) Deep reinforcement learning-based automatic exploration for navigation in unknown environment. IEEE transactions on neural networks and learning systems. pp. 2064-2076.

[13] Lin J, Yang X, Zheng P. (2019) End-to-end decentralized multi-robot navigation in unknown complex environments via deep reinforcement learning. In: 2019 IEEE International Conference on Mechatronics and Automation (ICMA), Tianjin. pp. 2493-2500.

[14] Zeng J, Ju R, Qin L. (2019) Navigation in unknown dynamic environments based on deep reinforcement learning.” Sensors. pp. 3837.

[15] Ye X, Lin Z, Li H. (2018) Active object perceiver: Recognition-guided policy learning for object searching on mobile robots,” In: 2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Madrid. pp. 6857-6863.

[16] Lv Y, Xie N, Shi Y. (2020) Improving target-driven visual navigation with attention on 3D spatial relationships. https://arxiv.org/pdf/2005.02153.

[17] Kulhánek J, Derner E, Babuška R. (2020) Visual Navigation in Real-World Indoor Environments Using End-to-End Deep Reinforcement Learning. https://arxiv.org/pdf/2010.10903.

[18] Devo A, Mezzetti G, Costante G. (2020) Towards generalization in target-driven visual navigation by using deep reinforcement learning,” IEEE Transactions on Robotics. pp. 1546-1561.

[19] Zhu Y, Mottaghi R, Kolve E. (2017) Target-driven visual navigation in indoor scenes using deep reinforcement learning. In: 2017 IEEE international conference on robotics and automation (ICRA), Marina Bay Sands. pp. 3357-3364.

[20] Klein G, Murray D. (2007) Parallel tracking and mapping for small AR workspaces. In: 2007 6th IEEE and ACM international symposium on mixed and augmented reality. Nara. pp. 225-234, 2007.

[21] Mur-Artal R, Montiel J M M, Tardos J D. (2015) ORB-SLAM: a versatile and accurate monocular SLAM system. IEEE transactions on robotics. pp. 1147-1163.

[22] Mur-Artal R, Tardos J D. (2017) Orb-slam2: An open-source slam system for monocular, stereo, and rgb-d cameras. IEEE Transactions on Robotics. pp. 1255-1262.
[23] EngelJ, SchoopsT, CremersD. (2014) LSD-SLAM: Large-scale direct monocular SLAM. In: European conference on computer vision. Springer. pp. 834-849.

[24] Mourikis A I, Roumeliotis S I. (2007) A multi-state constraint Kalman filter for vision-aided inertial navigation. In: 2007 IEEE International Conference on Robotics and Automation. Roma. pp. 3565-3572.

[25] Kim Y J, Elber G, Kim M S. (2014) Precise continuous contact motion for planar freeform geometric curves. Graphical models. pp. 580-592.

[26] Dorigo M, Birattari M, Stutzle T. (2006) Ant colony optimization. IEEE computational intelligence magazine. pp. 28-39, 2006.

[27] Anderson S J, Karumanchi S B, Iagnemma K. (2012) Constraint-based planning and control for safe, semi-autonomous operation of vehicles,” In: 2012 IEEE intelligent vehicles symposium. Alcalá de Henares. pp. 383-388.

[28] Anderson S J, Peters S C, Pilutti T E. (2010) An optimal-control-based framework for trajectory planning, threat assessment, and semi-autonomous control of passenger vehicles in hazard avoidance scenarios. International Journal of Vehicle Autonomous Systems. pp. 190-216.

[29] Pomerleau D A. (1989) Alvinn: An autonomous land vehicle in a neural network. CARNEGIE-MELLON UNIV PITTSBURGH PA ARTIFICIAL INTELLIGENCE AND PSYCHOLOGY PROJECT.

[30] Brahmbhatt S, Hays J. (2017) Deepnav: Learning to navigate large cities. In: IEEE conference on computer vision and pattern recognition. HI. pp. 5193-5202, 2017.

[31] Xu H, Gao Y, Yu F. (2017) End-to-end learning of driving models from large-scale video datasets. In: IEEE conference on computer vision and pattern recognition. HI. pp. 2174-2182.

[32] V. Mnih, A. P. Badia, M. Mirza, A. Graves, T. Lillicrap, T. Harley, D. Silver, and K. Kavukcuoglu. (2016) Asynchronous methods for deep reinforcement learning. In: International conference on machine learning. New York City. pp. 1928–1937.

[33] J. Schulman, F. Wolski, P. Dhariwal, A. Radford, and O. Klimov. (2017) Proximal policy optimization algorithms. http://arxiv.org/abs/1707.06347.

[34] Mirowski P, Pascanu R, Viola F. (2016) Learning to navigate in complex environments. https://arxiv.org/pdf/1611.03673.

[35] Luo Q, Sorokin M, Ha S. (2020) A Few Shot Adaptation of Visual Navigation Skills to New Observations using Meta-Learning. https://arxiv.org/pdf/2011.03609.

[36] Kolve E, Mottaghi R, Han W. (2017) AI2-thor: An interactive 3d environment for visual AI. https://arxiv.org/pdf/1712.05474.

[37] Mirowski P, Banki-Horvath A, Anderson K. (2019) The streetlearn environment and dataset. https://arxiv.org/pdf/1903.01292.

[38] Mo K, Li H, Lin Z. (2018) The adobeindoornav dataset: Towards deep reinforcement learning based real-world indoor robot visual navigation. https://arxiv.org/pdf/1802.08824.

[39] Qi C R, Litany O, He K. (2019) Deep hough voting for 3d object detection in point clouds. In IEEE/CVF International Conference on Computer Vision. Seoul. pp. 9277-9286.

[40] Anderson P, Chang A, Chaplot D S. (2018) On evaluation of embodied navigation agents. https://arxiv.org/pdf/1807.06757.