Experimental Research on Deep Reinforcement Learning in Autonomous navigation of Mobile Robot

Conference Paper · June 2019

DOI: 10.1109/ICIEA.2019.8833968

6 authors, including:

Yue Pengyu
Xi'an University of Technology
5 PUBLICATIONS **34** CITATIONS
SEE PROFILE

Xin Jing
Xi'an University of Technology
59 PUBLICATIONS **1,333** CITATIONS
SEE PROFILE

Ding Liu
Xidian University
159 PUBLICATIONS **2,681** CITATIONS
SEE PROFILE

Mao Shan
The University of Sydney
77 PUBLICATIONS **1,008** CITATIONS
SEE PROFILE

Experimental Research on Deep Reinforcement Learning in Autonomous navigation of Mobile Robot

Pengyu Yue [1,2] , Jing Xin [1,2,*],Huan Zhao[1],Ding Liu[1]
1. Xi'an University of Technology
2. Shaanxi Key Laboratory of Integrated and Intelligent Navigation, CETC 20
Xi'an, China
*corresponding author: xinj@xaut.edu.cn

Mao Shan
Australian Centre for Field Robotics
The University of Sydney
Sydney, Australia
m.shan@acfr.usyd.edu.au

Jian Zhang
Global Big Data Technologies Centre
University of Technology Sydney
Sydney, Australia
Jian.Zhang@uts.edu.au

***Abstract*—The paper is concerned with the autonomous navigation of mobile robot from the current position to the desired position only using the current visual observation, without the environment map built beforehand. Under the framework of deep reinforcement learning, the Deep Q Network (DQN) is used to achieve the mapping from the original image to the optimal action of the mobile robot. Reinforcement learning requires a large number of training examples, which is difficult to directly be applied in a real robot navigation scenario. To solve the problem, the DQN is firstly trained in the Gazebo simulation environment, followed by the application of the well-trained DQN in the real mobile robot navigation scenario. Both simulation and real-world experiments have been conducted to validate the proposed approach. The experimental results of mobile robot autonomous navigation in the Gazebo simulation environment show that the trained DQN can approximate the state action value function of the mobile robot and perform accurate mapping from the current original image to the optimal action of the mobile robot. The experimental results in real indoor scenes demonstrate that the DQN trained in the simulated environment can work in the real indoor environment, and the mobile robot can also avoid obstacles and reach the target location even with dynamics and the presence of interference in the environment. It is therefore an effective and environmentally adaptable autonomous navigation method for mobile robots in an unknown environment.**

Keywords—Deep reinforcement learning, DQN, mobile robot, nature scene, autonomous navigation

I. Introduction

Recently, autonomous navigation technology has been extensively used in various unmanned systems such as unmanned vehicles, drones, unmanned boats, and so on. As the range of applications for robots continues to expand, the environment in which robots are intended to operate is becoming increasingly complex [1]. To autonomously navigate through an environment and perform a variety of tasks, many mobile robot systems heavily rely on the map and their own location information. The robot needs to reason the next action based on the known information. However, due to the limited environmental information accessible to robots and the unpredictability of environmental conditions, the practical application of conventional navigation algorithms is restricted [2].

Reinforcement learning (RL) techniques can learn appropriate actions from the environment states. In the process of interaction between the agent and the external environment, the agent repeatedly learns through trial and error, obtains environmental information, and continuously optimizes the action strategy of the agent [3]. This optimization method gives the RL an excellent decision-making ability [4]. At present, reinforcement learning has been successfully applied in mobile robot path planning [3]. However, the performance of RL usually heavily depends on the selection of artificial features. The quality of features selected significantly affects the learning outcomes [5]. In order to solve this problem, Google's artificial intelligence research team DeepMind combines the strong perception of deep learning with the excellent decision-making ability of reinforcement learning [5-8], and successfully applies it to Atari video games and computer Go. The algorithm uses the game screen as information input and the game score as reward information. Finally, satisfactory agent control performance is achieved. Deep learning reinforcement learning will play an increasingly important role in achieving artificial general intelligence [9-11].

Although gaming is different from the robot control, the mechanism of the agent learning the action strategy in the given environment is similar when it comes to the mobile robot to achieve the path planning only using the current visual perception. Inspired by this, the deep reinforcement learning technology is applied in the paper to the autonomous navigation of mobile robots, that is, the collision-free motion from the current location to the desired location (target object) using only the original visual information.

The remainder of the paper is organized as follows. The overall framework and main components of the proposed method are described in Section II, followed by experimental validation in Section III. Lastly, the conclusion is drawn in Section IV.

II. PROPOSED METHOD

A. The overall framework of the proposed planning method

The overall framework of the deep reinforcement learning method based mobile robot autonomous navigation approach proposed in this paper is shown in Fig. 1. The framework consists of a mobile agent (robot) and a given environment where the agent operates. The agent part mainly includes three components: 1) image acquisition and preprocessing, 2) value function acquisition, and 3) action selection. The goal of the reinforcement learning is to maximize the cumulative reward value of mobile robots received from the environment. In the current research, RL has demonstrated its decision-making ability beyond humans in strategic games [12-13], computer Go [8-9], manipulator control[14-15],and autonomous driving [3,16]. However, unlike in the simulation environment, it is difficult to obtain reward information in the physical environment. Even if

978-1-5386-9490-9/19/$31.00 ©2019 IEEE

obtainable in the physical environment, the reward obtained is often accompanied with noise and reward delay. Therefore, most of existing reinforcement learning applications are only achieved in the simulation environments or games.

In the paper, we propose to use a two-step way to obtain optimal action strategy model. Firstly, the DQN model is trained in the simulation environment. The well-trained DQN is then applied to the real mobile robot autonomous navigation scene. The corresponding simulation platform and experimental platform are constructed in the Gazebo simulation environment and the real indoor environment respectively. In the robotics system, the inputs of the DQN is the current RGB image captured in real time, and the network output is the Q value corresponding to each possible action of the robot. The mobile robot selects the optimal action using the action selection strategy to avoid the obstacle and eventually reach the target position.

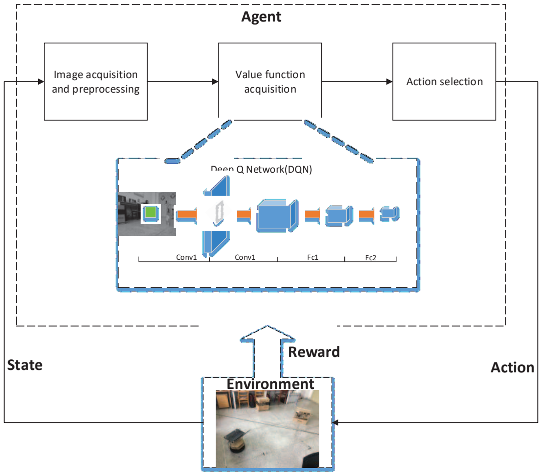


Fig.1 Overall block diagram of autonomous navigation process of mobile robot based on deep reinforcement learning

B. Module function and implementation

In this section, we will focus on the specific implementation process of the proposed approach, which can be decomposed to image acquisition and preprocessing, value function acquisition, and action selection.

(1) Image acquisition and preprocessing

The main function of the module is to reduce the computational burden of subsequent image processing. Firstly, the image is grayed, and the dimensionality reduction is then performed using down-sampling. Finally, the latest 4 frames of preprocessed images are stacked up as the current environmental state. The flow chart of the image acquisition and preprocessing is illustrated in Fig. 2.

(2) Value function acquisition

The main function of the value function acquisition module is to use the DQN to compute the state-action value function Q. The input of the DQN is the most recent 4 frames of preprocessed images, and the output of the DQN is the Q value of each possible action of the robot.

During the training process, the bonus function used is presented in Equation (1):

$$r=\begin{cases}-50 & d_{obs}<d_1, collision\\ d_{tar}(t-1)-d_{tar}(t) & otherwise\\ 50 & d_{tar}<d_2, d_{tar}=0\end{cases} \qquad (1)$$

where, d_{obs} is the distance from the robot to the obstacle in meters. $d_{tar}(t)$ is the distance between the mobile robot and the target point at the current moment t . $d_{tar}(t-1)$ is the distance between the mobile robot and the target point at the last time $t-1$. d_1 is the collision distance threshold (unit: m). $d_{obs}<d_1$ indicates collision of the robot with the obstacle. d_2 is the target threshold distance, and $d_{tar}<d_2$ indicates that the robot has reached the target object. These distance values can be trivially provided in the simulation environment.

The detailed training process of DQN is listed as follows:
Step 1: Initialize the learning rate, attenuation factor γ, experience memory replay_memory_size , sample size batch_size, ε start value and end value and other parameters;
Step 2: Initialize the DQN with random weights w;
Step 3: Set the number of iterations and for each training iteration:
a) Obtain the initial state image and perform the image preprocessing;
b) Determine whether the termination condition is satisfied, and if so, exit the loop, otherwise, perform the following steps:
i) Select the action using the ε-greedy strategy and perform the action;
ii) Update the state information and reward information after the action is performed;
iii) Store the <current state, action, reward, next state> into the experience memory, and randomly select the training sample using the experience replay mechanism.
iv) Update the network weights using stochastic gradient descent.

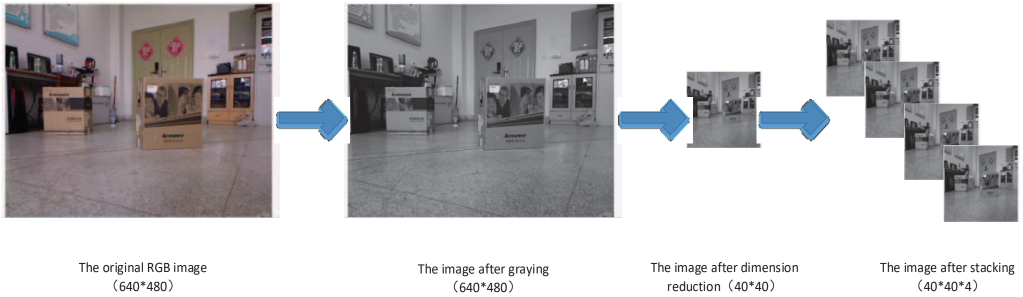


Fig.2. Image acquisition and preprocessing

In the test process of the DQN, the environment no longer provides reward information. The value function directly uses the DQN network to obtain the state-action value function Q (s, a). The input of the DQN network is the preprocessed latest 4 frames image using the environment perception module. The output of the DQN network is the Q value corresponding to each action of the mobile robot.

The navigation process of the mobile robot using DQN is detailed as follows:

Step 1: Obtain the environment state information s from the current image captured by the robotics system, and let the current state $s = s_0$;
Step 2: Adopt the well-trained DQN to select the current optimal action with the largest Q value corresponding to the current state, that is, the optimal action a satisfying $Q(s,a) = \max_{a'} Q(s,a')$;
Step 3: Perform the optimal action a, and get the next moment state s';
Step 4: Update the current state, $s = s'$ and determine if there is an object in the range of d_1, if any, stop the action, and otherwise repeat step 2 and step 3.

The DQN network architecture designed in this paper is shown in Table 1. Two convolutional layers (Convl, Conv2) are used for image feature extraction, and two fully connected layers (fc1, fc2) are used for policy learning. In order to increase the degree of nonlinearity, both of the convolutional layer and the fully connected layer fc1 use a modified linear unit (Relu) activation function, and fc2 outputs a corresponding Q (a, s) using a linear function. The step size is 2 in convolutional layers.

TABLE I. DQN NETWORK ARCHITECTURE

Floor	*Input*	*Convolution kernel size*	*Number of features*	*Output*
Conv1	40*40*4	3*3	8	19*19*8
Conv2	19*19*8	3*3	16	9*9*16
Fc1	9*9*16	/	128	128
Fc2	128	/	Actions number	Actions number

(3) Action selection

The main function of the action selection module is to determine the optimal action of the mobile robot using the action selection strategy. In the paper, action selection strategy adopts the popular ε-greedy strategy [15]. Mobile robot will perform random action selection with probability ε in the current state to ensure the degree of exploration of the state space, and choose the action with probability 1-ε that maximizes the current Q value to exploit the learned knowledge as much as possible. In the training process, the value of ε can be computed by Equation (2).

$$\varepsilon_{t+1} = \varepsilon_t - \frac{\varepsilon_{initial} - \varepsilon_{end}}{Explore_{steps}} \tag{2}$$

Where $\varepsilon_{initial}$ and ε_{end} are the initial value and the final value of ε respectively, ε_t is the value of ε at the current time, and $Explore_{steps}$ is the number of step for exploring, which can be preset.

The possible actions a of the mobile robot are turning left, going straight, and turning right, as shown in equation (3), which can be achieved by different line speeds v and angular velocity ω.

$$a = \begin{cases} v = 0.05m/s, \omega = 0.5rad/s & turn\ left \\ v = 0.2m/s, \omega = 0rad/s & go\ straight \\ v = 0.05m/s, \omega = -0.5rad/s & turn\ right \end{cases} \tag{3}$$

III. EXPERIMENTAL RESULTS

In order to verify the effectiveness of the deep reinforcement learning based mobile robot autonomous navigation method proposed in this paper, two autonomous navigation experiments are conducted in the Gazebo simulation environment and a real environment respectively.

A. Mobile robot autonomous navigation in Gazebo simulation environment

Deep reinforcement learning requires a significant number of instances where the agent interacts with the environment for the learning of the mapping from the current image to an optimal action of mobile robot. Hence, we first train the DQN under the Gazebo simulation platform for the purpose of reducing the training time and the cost of the experiment.

The constructed simulation navigation environment using the Gazebo platform is shown in Fig. 3. The left view in Fig. 3 is the current local image obtained from the first-person perspective of the mobile robot, and the right view is global image obtained from the third-person perspective for observation. The target object in the navigation is a Coke can, and the Turtlebot mobile robot is trained to complete the autonomous navigation task. The entire moving environment is a 6*8m area surrounded by walls. The length, width and height of two identical obstacles are 0.5m, 0.4m and 0.5m respectively. The starting position of the Turtlebot robot is (0m,0m); The position of the target object (Coke can) is (-2m, -3m), and the positions of the obstacles (boxes) are (0m, 2m) and (-1m, -1m); the tables are around the robot and the obstacles.

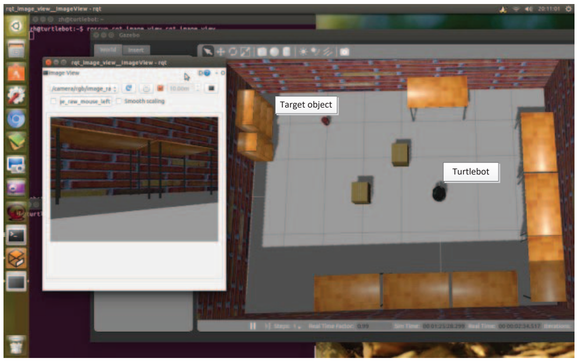


Fig. 3. Gazebo simulation environment diagram

During the training, the information obtained by the mobile robot includes the images provided by the Kinect V1 and the reward information provided by the Gazebo platform. During the training, the Turtlebot robot interacted with the environment shown in Fig. 3 for a total of 50,000 trials (Episode_num). The termination conditions for the learning process are: 1) reaching the target object position, 2) colliding with any obstacle, or 3) reaching the maximum number of steps (Max_steps) per experiment.

The experimental parameters in the training process are shown in Table 2:

TABLE II. TRAINING PROCESS PARAMETERS

parameter	***value***
γ	0.99
Initial_ ε	1.0
End_ ε	0.1
Explore_steps	200,000
Observe_steps	640
Replay_memory_size	1,000,000
Batch_size	64
Episode_num	500,000
Max_steps	1000
d_1	0.5
d_2	0.35
Learning_rate	0.000001

After the training, the autonomous navigation ability of the Turtlebot is tested in the Gazebo environment. The navigation results are shown in Fig. 4.

Simulated navigation experimental results show that the proposed navigation strategy can successfully achieve autonomous and collision-free motion of the robot to the location of the target object without building the environment map in advance. It is an effective autonomous navigation scheme, which verifies the feasibility of the deep reinforcement learning in the mobile robot autonomous navigation.

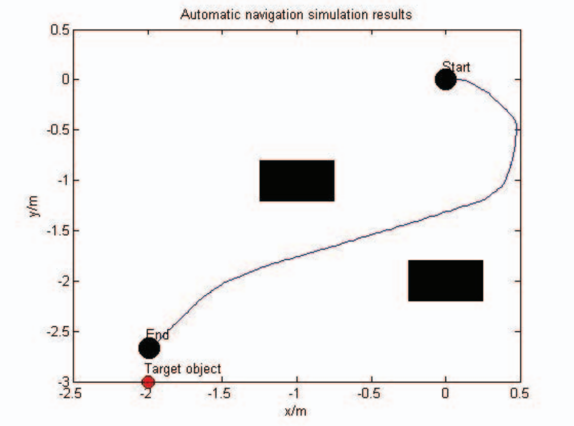


Fig. 4. Autonomous navigation simulation results

B. *Mobile robot autonomous navigation in the real nature indoor environment*

The purpose of the real-world experiment is to further validate the proposed navigation strategy in the real indoor environment.

The shape and appearance of the navigation target object is shown in Fig. 5. Mobile robot autonomous navigation experimental platform is shown in Fig.6. The experimental platform includes hardware part and software part. The hardware part consists of the Turtlebot2 mobile robot, the Kinect V1 sensor, a laptop, and a target object. The software part is robot operating system ROS (ROS Kinetic) in the Ubuntu 16.04 environment. In the experiment, the Kinect V1 acquires the image data and transmits it to the laptop computer, and then the computer processes the data to control the Kobuki robot base to perform the action. The well-trained DQN network in the Gazebo simulation platform is used to achieve the mapping from the current image captured by navigation system shown in Fig. 6 to the optimal action that Turtlebot2 mobile robot is to perform.

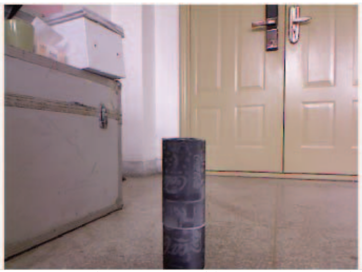


Fig. 5. Target object

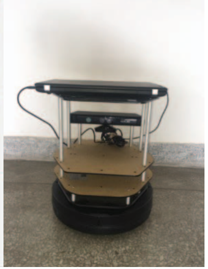


Fig. 6. Turtlebot2 mobile robot

Fig. 7 shows the physical experiment environment. The left image shows the image obtained by the first person perspective of the mobile robot. The right image shows the observer's perspective.

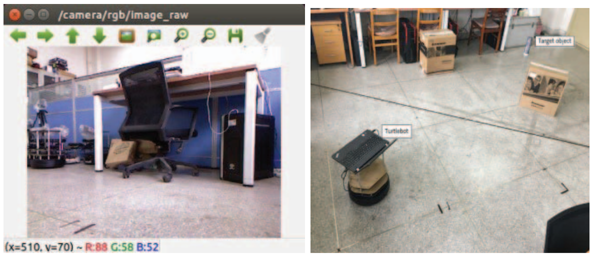


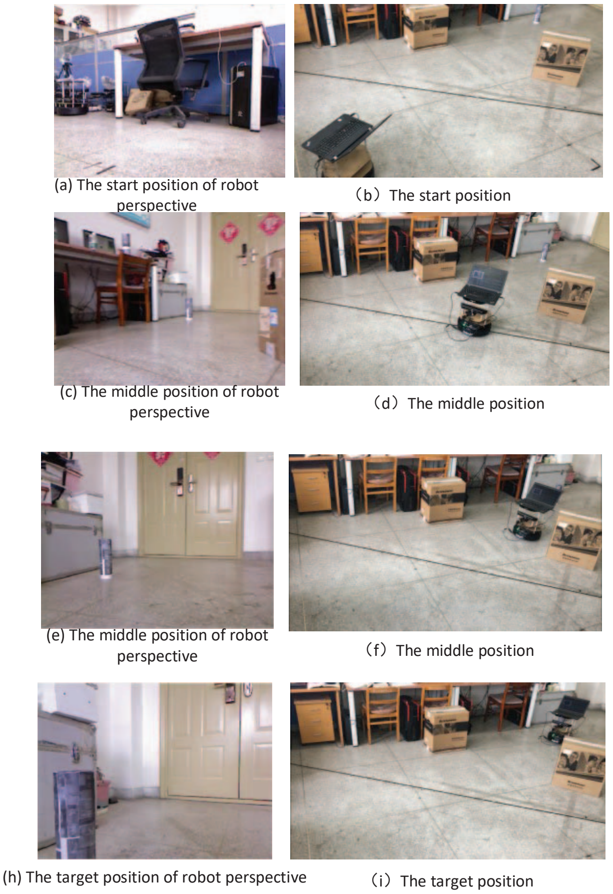
Fig. 7. Physical environment

The configuration of experimental parameters is fine-tuned relative to the training process, as shown in Table 3.

TABLE III. TEST PROCESS PARAMETERS

parameter	value
γ	0.99
End_ ε	0
Max_steps	9000
d_1	0.35

Turtlebot's movement in the physical environment is shown in Fig. 8. Fig. 8(a), (c), (e), (h) are images taken by the on-board camera of Turtlebot at 7 different times. The pictures are separated from one another by 23s. Fig. 8(b), (d), (f) and (i) are top views of the environment at the corresponding moments of Fig. 8(a), (c), (e), (h), respectively. The above experimental results show that the artificial intelligence navigation algorithm based on deep reinforcement learning enables the mobile robot to learn independently only from visual observations in an unfamiliar environment and complete the collision-free movement from the initial position to the target location. Once again, the validity and feasibility of the proposed algorithm in the mobile robot autonomous navigation are verified.



(a) The start position of robot perspective

（b）The start position

(c) The middle position of robot perspective

（d）The middle position

(e) The middle position of robot perspective

（f）The middle position

(h) The target position of robot perspective

（i）The target position

Fig.8. Turtlebot motion process

IV. Conclusion

This paper proposes a mobile robot autonomous navigation algorithm based on deep reinforcement learning. Navigation experimental results show that the proposed navigation algorithm can enable the real robot to perform autonomous and collision-free movement to the target position without building the environment map in advance. It can also be concluded from the experimental results that the proposed approach is an effective autonomous navigation method with strong environmental adaptability.

Acknowledgment

This work is supported by the Chinese National Natural Science Foundation under Grant No. 61873200 and Open project of Shaanxi Key Laboratory of Integrated and Intelligent Navigation under Grant No. SKLIIN-20180205.

References

[1] T.Lei , G. Paolo , and M. Liu . "Virtual-to-real Deep Reinforcement Learning: Continuous Control of Mobile Robots for Mapless Navigation.". 2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS),pp. 31-36.

[2] J. Xin, H. Zhao, D. Liu, et al. "Application of deep reinforcement learning in mobile robot path planning". 2017 Chinese Automation Congress & China Intelligent Manufactory International Conference (CAC 2017 & CIMIC). Jinan,China,2017, pp.7112-7116

[3] R. S. Sutton , and A. G. Barto . "Reinforcement Learning: An Introduction *(2nd Edition, in preparation)* ". Massachusetts: MIT Press. 2017

[4] S.Ahmadel,A.Mohammed,P.Etienne, et al. "Deep Reinforcement Learning framework for Autonomous Driving".Electronic Imaging, 2017(19): 70-76.

[5] L.W. Qiu, “Application of Deep Reinforcement Learning In Video Game Playing,” Master Thesis of South China University of Technology,(2015)

[6] V. Mnih, K. Kavukcuoglu, D. Silver, et al. “Playing Atari with Deep Reinforcement Learning”, In Deep Learning, Neural Information Processing Systems Workshop, 2013.

[7] V. Mnih, K. Kavukcuoglu, D. Silver, et al. "Human-level control through deep reinforcement learning." Nature,2015,518(7540):529-533.

[8] D. Silver, A. Huang, C.J. Maddison, et al. "Mastering the game of Go with deep neural networks and tree search". Nature, 2016, 529(7587): 484-489.

[9] D. Silver, J.Schrittwiesrh, K. Simonyan, et al. "Mastering the game of Go without human knowledge". Nature, 2017,550(7676): 354 - 359

[10] Li, Yuxi . "Deep Reinforcement Learning: An Overview" . 2017, arXiv preprint arXiv: 1701.07274

[11] Z.Tang, K.Shao,D.Zhao, et al. "Review of deep reinforcement learning and discussions on the development of computer Go" .Control Theory & Applications,2017,34(12):1529-1546

[12] J.Schulman,F.Wolski,P.Dhariwal, et al. "Proximal policyoptimization algorithms". arXiv preprint arXiv: 1707.06347,2017

[13] L.Guillaume , and D. S. Chaplot . "Playing FPS Games with Deep Reinforcement Learning". The Thirty-First AAAI Conference on Artificial Intelligence (AAAI2017), San Francisco, California, USA,pp. 2140-2146

[14] S.Levine, C.Finn, T.Darrell, et al. "End-to-end training of deep visuomotor policies". Journal of Machine Learning Research,2016, 17(39): 1-40

[15] S.Gu, T.Lillicrap, I.Sutskever, et al. "Continuous deep Qlearning with model-based acceleration". The 33rd International Conference on Machine Learning (ICML).2016, New York,PP.2829 -2838.

[16] Y.Zhu, R.Mottaghi, E.Kolve, et al. "Target-driven visual navigation in indoor scenes using deep reinforcement learning". 2017 IEEE International Conference on Robotics and Automation (ICRA), Singapore, pp. 3357-3364.