Intelligent Service Robotics (2020) 13:263–272
https://doi.org/10.1007/s11370-019-00310-w

ORIGINAL RESEARCH

Path planning for active SLAM based on deep reinforcement learning under unknown environments

Shuhuan Wen[1] · **Yanfang Zhao**[1] · **Xiao Yuan**[1] · **Zongtao Wang**[1] · **Dan Zhang**[2] · **Luigi Manfredi**[3]

Received: 24 December 2018 / Accepted: 25 December 2019 / Published online: 16 January 2020
© Springer-Verlag GmbH Germany, part of Springer Nature 2020

Abstract

Autonomous navigation in complex environment is an important requirement for the design of a robot. Active SLAM (simultaneous localization and mapping) combining, which combine path planning with SLAM, is proposed to improve the ability of autonomous navigation in complex environment. In this paper, fully convolutional residual networks are used to recognize the obstacles to get depth image. The avoidance obstacle path is planned by Dueling DQN algorithm in the robot's navigation; at the same time, the 2D map of the environment is built based on FastSLAM. The experiments show that the proposed algorithm can successfully identify and avoid moving and static obstacles with different quantities in the environment, and realize the autonomous navigation of the robot in a complex environment.

Keywords Path planning · FastSLAM · Deep reinforcement learning

1 Introduction

Mobile robots can navigate autonomously in an unknown environment. This task needs faces three main subproblems: (i) localization, (ii) mapping and (iii) planning. Simultaneous localization and mapping (SLAM) has been well studied in the past decades. Research institutes have proposed several solutions to the path planning problem. Nowadays, mobile robots can navigate autonomously from a starting point A to an ending point B. They can interact with the environment and take appropriate action in different scenarios [1, 2].

However, path planning combined with SLAM is still in its early stages [3]. A robot can follow a pre-programmed path autonomously, or with a user manual control relying on the traditional SLAM method [4]. In the movement process of the mobile robot, real-time data of the scene obtained from its own sensors are used for localization. A designed walking path is manually set, which is not the autonomous navigation for the robot. This approach limits the autonomy and initiative of the robot motion, including the online path planning function.

On the other hand, the main goal of path planning is obstacle avoidance [5]. In an unknown environment, the path of an obstacle is also unknown [6], which implies mobile robots' cooperation on all subtasks (including localizes,

Electronic supplementary material The online version of this article (https://doi.org/10.1007/s11370-019-00310-w) contains supplementary material, which is available to authorized users.

✉ Shuhuan Wen
swen@ysu.edu.cn

Yanfang Zhao
fangyzhao@163.com

Xiao Yuan
yuanxiaohb@163.com

Zongtao Wang
732023424@qq.com

Dan Zhang
dan.zhang@lassonde.yorku.ca

Luigi Manfredi
mail@luigimanfredi.com

[1] Key Lab of Industrial Computer Control Engineering Hebei Province, Yanshan University, Qinhuangdao 066004, China

[2] Department of Mechanical Engineering, Lassonde School of Engineering, York University, Toronto, Canada

[3] Institute for Medical Science and Technology (IMSaT), University of Dundee, Dundee, UK

maps and plans). This allows robots to handle the real environment without any external supervision.

This paper focuses on improving the autonomous learning ability and adaptability of a robot (Turtlebot2) in an unknown environment by using active SLAM. This allows integration of path planning into the framework. We propose a path planning algorithm based on deep reinforcement learning algorithm (DRL) in the SLAM formulation in an unknown environment. The deep reinforcement learning algorithm learns the interaction between agent and environment and solves the problem of the path planning of mobile robot. In SLAM, a robot uses its own sensors to identify signatures from an unknown environment and then estimates the robot's global coordinates and signatures based on the relative position of the robot and the odometer data [7]. The online positioning and map building require maintaining detailed information between the robot and the functional identity [8].

2 Related work

In recent years, SLAM research has made great progress, applying these results to indoor, underwater and outdoor environments [3]. Some approaches do not consider path in the objective function, and others focus on reducing uncertainty in order to perform efficient exploration [9, 10]. In this paper, both path planning using DRL and robot navigation uncertainty are investigated.

Deep learning has shown its advantage in robotics and computer vision. Especially, the path planning learning based on deep learning to avoid collisions has become increasingly popular. Predicting control strategies directly from the original image [11–13], which can avoid the complex modelling and parameter adjustment of traditional path planners, are particularly popular. Deep learning is the inherent law and presentation level of learning sample data, and the information obtained in the learning process is helpful to the interpretation of data such as words, images and sounds [14, 15]. Its ultimate goal is to make machines as capable of analysing and learning as people are and also be able to recognize data. In essence, deep learning aims to build a machine learning architecture model with multiple hidden layers [16]. Through training on large-scale data, a large amount of representative characteristic information can be obtained. Therefore, the sample is classified and predicted to improve the accuracy of classification and prediction. The process aims to achieve feature learning through deep learning model. Although deep learning is a feasible approach to benefit from large datasets, the learnt policy is essentially limited by the label generating strategy. Reinforcement learning (RL), though, is good at controlling an individual who can act autonomously in a certain environment and constantly improve its behaviour through the interaction with the environment [17]. The reinforcement learning problem involves learning for mapping the environment and then translating that into action to maximize the reward. In reinforcement learning, a learning machine is a decision-making agent without any external control on the action it needs to perform. This algorithm repeatedly runs a process to discover the behaviour to the maximum reward. In general, actions affect not only the current rewards, but the environment in the next point, and therefore affect all subsequent rewards. Because the learning system will affect the environment, this environment will affect the follow-up action. Deep reinforcement learning combines the perceptive ability of deep learning with the decision-making ability of reinforcement learning in a general form, and learns a mapping from the original input to the action output, through to an end-to-end approach [14]. In many large-scale decision-making tasks based on visual perception, the deep reinforcement learning method has already made a breakthrough. We use visual information to transfer the RGB image to RGB-D image to infer walking space and the obstacle space, to avoid obstacles by path planning algorithm based on DRL.

In this paper, fully convolutional residual network (FCRN)is used to recognize the obstacles to get depth image and build the actual environment 2D maps. An active SLAM method integrating the obstacle avoidance algorithm based on deep reinforcement learning into the SLAM framework is proposed. The avoidance path of the obstacle is planned by Dueling DQN algorithm in the robot's navigation; at the same time, the 2D map of the environment is built based on FastSLAM. Experiments demonstrate that the proposed method can effectively avoid obstacles in the environment with different numbers of static and dynamic obstacles and realize the autonomous navigation of the robot in the complex environment.

We adopt our previous active SLAM framework combining reinforcement learning (RL) path planning with EKF-SLAM [18]. Further work is studied in this paper. We use a deep reinforcement learning [19] to avoid the dynamic and static obstacles based on FastSLAM. The robot will encounter obstacles with different quantity and state when it navigates in complex environment. The robot can avoid dynamic and static obstacles using the deep reinforcement learning algorithm and build the map of the unknown environment successfully. Dueling DQN (D3QN) is more efficient than traditional deep Q-network (DQN) [19]. So D3QN is used to plan the path of the avoidance obstacles in this paper.

We use our previous active SLAM framework [18] to further improve the ability of autonomous navigation for the robot. The contributions of our work are as follows: FCRN is used to obtain the depth image prediction, and then D3QN algorithm and FastSLAM are used to avoid obstacles and map the environment, which efficiently finishes

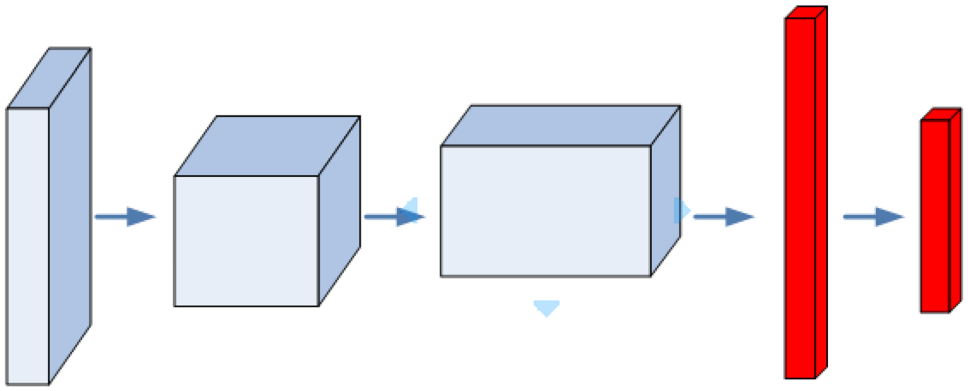

Fig. 1 Q-network structure of a traditional DQN learning algorithm

the autonomous navigation for the robot in complex environment. Different experimental scene is used to test the ability of autonomous navigation, including dynamic and static obstacles in the environment. Two static obstacles, four static obstacles, two static obstacles and one dynamic obstacle are used to verify the effectiveness of the proposed method. The experiments show that the proposed method can successfully identify and avoid the obstacles with different types and quantity in the environment, and realize the map construction of mobile robot in the unknown environment. For traditional SLAM, the robot can only walk along the pre-programmed path, which limits the ability of autonomous navigation of the robot in the complex environment. However, the proposed active FastSLAM based on deep reinforcement learning solves the problem of traditional SLAM that lacks the ability of the autonomous navigation in an unknown environment and is not able to build an environmental map efficiently.

3 Deep reinforcement learning

Aggrandizement learning agents have achieved some success in some fields, and their applicability has previously been limited to areas where manual production has useful characteristics, or fields with fully observed, low-dimensional state space [19]. In order to successfully use reinforcement learning in close proximity to real-world complexity, agents face a difficult task. They must obtain effective representations of the environment from high-dimensional

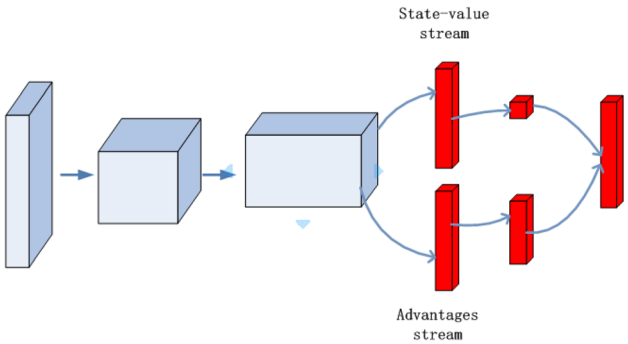

Fig. 2 Dueling network has two streams to separately estimate state value and the advantages for each action

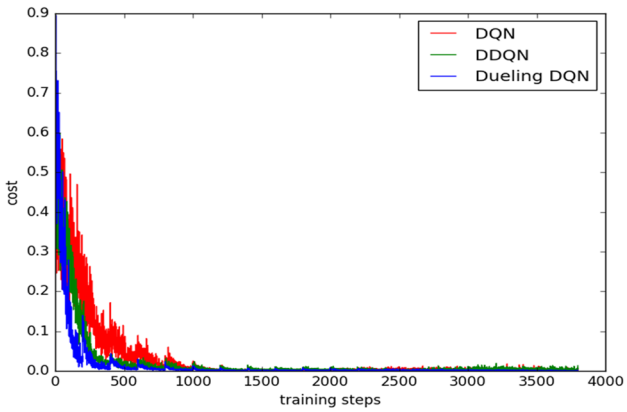

Fig. 3 Loss comparison of DQN, DDQN and Dueling DQN

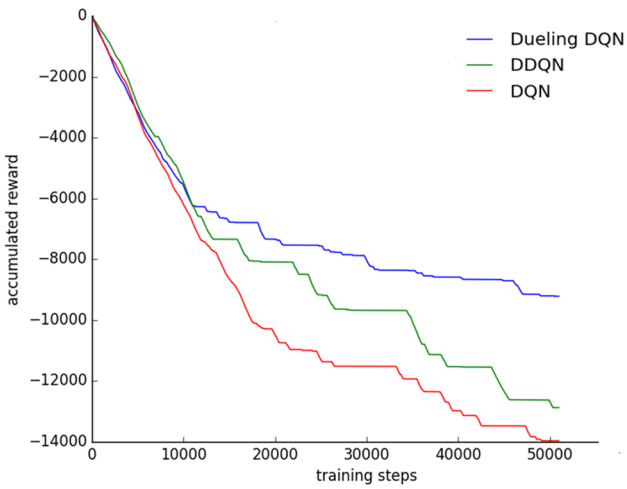

Fig. 4 Learning curves of DQN, DDQN, Dueling DQN with average rewards acquired by robot

sensory inputs and use this information to generalize past experiences into new situations. We use the latest advances in training deep neural networks to develop a new artificial agent, called deep Q-network. It can learn successful strategies directly from high-dimensional sensor input through end-to-end intensive learning.

In a traditional Q-learning, Q-table can be used to store the Q value of each state action pair when the state and action space are discrete, and the dimension is not high, while it is unrealistic to use Q-table when the state and action space are high-dimensional continuous.

The usual approach is to turn the update problem of Q-table into a function fitting problem and the similar state results in similar output actions. By updating the parameter, the Q function approximates the optimal Q value. In Eq. (1), s is the state, a is the action, and θ is the parameter:

$$Q(s, a;\theta) \approx Q'(s, a) \tag{1}$$

The deep neural network can automatically extract complex features, so it is most suitable to use the deep neural network in the face of high-dimensional and continuous state [13]. Dueling DQN improves the network architecture. It uses the model structure to represent the value function in a more detailed form, which enables the model to have higher performance and reduces the overestimation of the Q value of DQN.

Different from the traditional DQN structure, as shown in Fig. 1, the dueling network structure is composed of two flows: one is the state value estimation, and the other is the state-independent motion advantage function. The underlying convolution feature learning module is shown in Fig. 2. A major benefit of this decomposition is that it generalizes

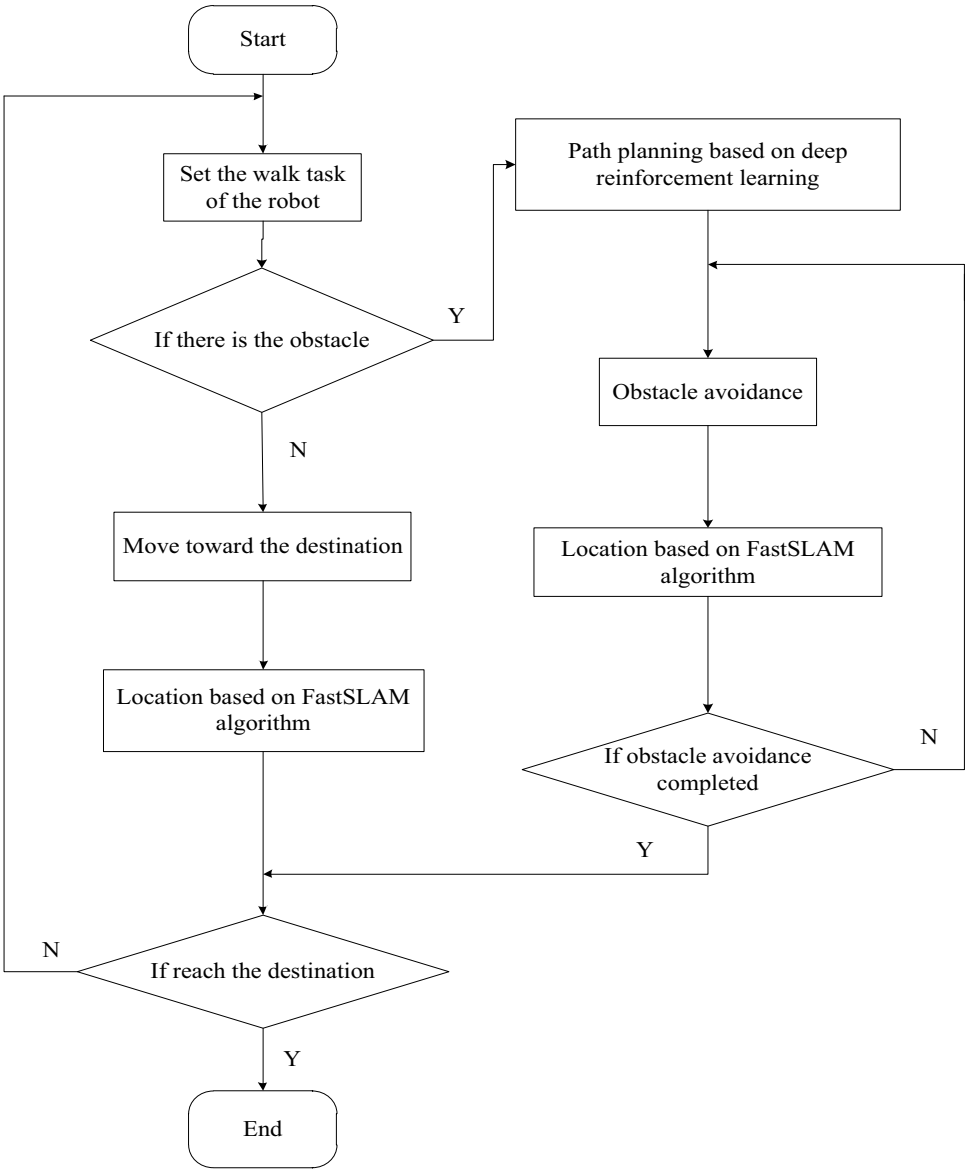

Fig. 5 Flow chart of experiment

the learning between actions without changing the reinforcement learning algorithm. The dueling network should also be understood as a single Q-network, with two streams instead of one. Intuitively, the dueling network can learn what states have or do not have value without learning the effects of each action in each state. This is especially useful specifically for states where actions do not affect the environment [20]. We compared the training efficiency and performance of Dueling DQN (D3QN), deep double Q-network (DDQN) and deep Q-network (DQN). The comparison result is shown in Figs. 3 and 4. Figure 3 shows that the loss of D3QN is the fastest, DDQN is the second, and the loss of DQN is the slowest, which demonstrates that D3QN is better than DDQN, and DDQN is better than DQN. Figure 4 shows that D3QN model outperforms the other two models. The learning strategy with D3QN model structure is more rewarding. As the training time increases, the cumulative reward of D3QN becomes higher. This paper reports a D3QN model to plan the path to avoid obstacles.

Value function Q is decomposed into state value function (V) and advantage function (A), in formula (2):

$$Q^{\pi}(s,a) = V^{\pi}(s) + A^{\pi}(s,a) \tag{2}$$

The advantage function shows the difference between current action and average performance. If this value is better than average, then the advantage function is positive, the opposite if this value is negative [21]. We add a limit to the advantage function as we know that the expectation of the advantage function is 0. Equation (2) becomes the following one:

$$Q^{\pi}(s,a) = V^{\pi}(s) + (A^{\pi}(s,a) - \frac{1}{|A|}\sum_{a'} A(s,a')) \tag{3}$$

By subtracting the average value from each A value, the constraint that the expected value is 0 can be guaranteed, which in turn increases the stability of the collected output. If the ordinal uses the value of V in some scenarios, we do not need to train a network. At the same time, by explicitly giving the output value of V function, we explicitly update V function every time we update it [22]. In this way, the update frequency of V function will be increased with certainty. For single output Q-network, its update is somewhat obscure. From the perspective of network training, this makes network training more user friendly and easy.

4 Path planning based on FastSLAM (active SLAM)

The robot navigates in an indoor environment and builds the map of the environment. If the robot detects an obstacle, it will plan a path to avoid the obstacle by using deep reinforcement learning before reaching the minimum distance between the robot and the obstacle. Otherwise, the robot will continue to navigate until it finally reaches the destination. The flow chart of experiment is shown in Fig. 5.

4.1 Action space representation

In an indoor environment, the robot motion forms include the linear velocity and the angular velocity. The behaviour of

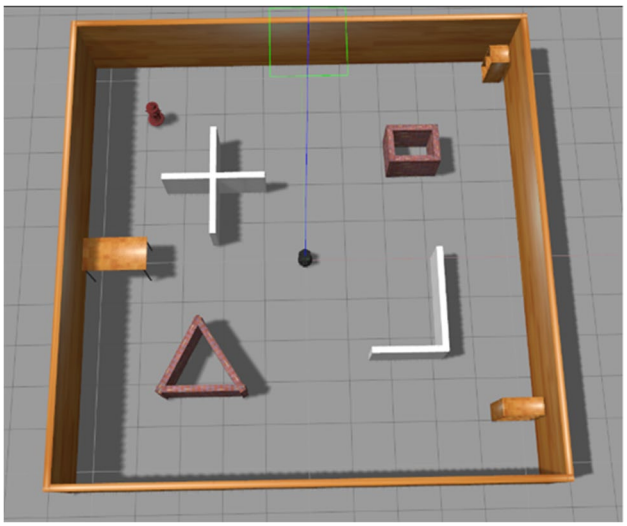

Fig. 6 Simulation environment of Gazebo [12]

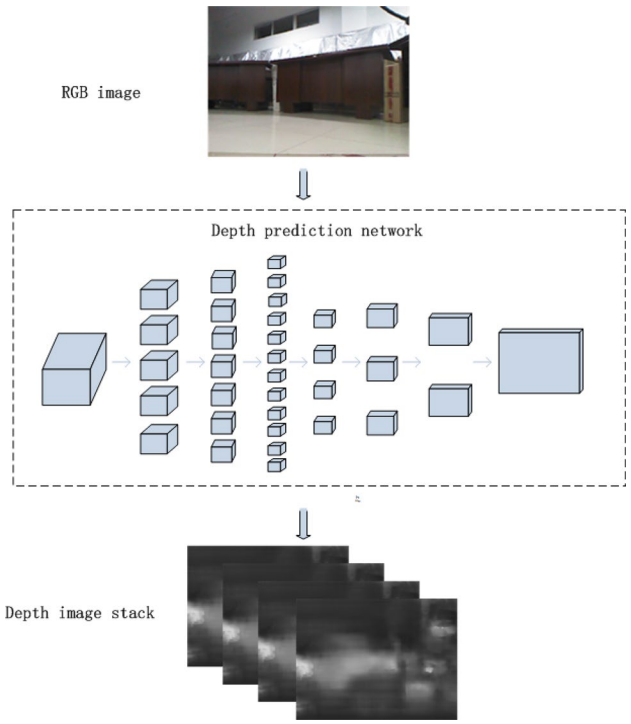

Fig. 7 Static image motion prediction training network structure

a robot consists of ten basic operations. The linear velocity is set to be 0.4 or 0.2 m/s, while the angular velocity is $\frac{\pi}{6}$, $\frac{\pi}{12}$, 0, $-\frac{\pi}{12}$, $-\frac{\pi}{6}$ rad/s, producing ten different behaviours. These ten actions can not only help the robot effectively complete the path planning task, but also improve the operational efficiency of deep reinforcement learning algorithm. When the robot encounters obstacles, it is punished. When the robot reaches its goal, it is rewarded. Only with this strategy, the robot can gradually approach the goal and, at the same time, remove the obstacles in the learning process to finally complete the path planning task.

4.2 Reward function

Reward function is defined by the external environment and the subjective objects. The merits of the reward function design play an important role in the learning speed and quality. In order to train the network to produce a feasible control strategy, it is necessary to correctly define the actions of the robot. Unlike simple commands such as "continue" or "turn left or right", the actions in our network are defined to control line speed and angular velocity in discrete formats. The instantaneous reward function is defined as $r = v * \cos(\omega) * \mathrm{dt}$ and ω, respectively, local line speed and angular velocity, and dt is the second of each training time loop 0.2. The bonus function is to let the robot run as fast as possible and punish the robot with a simple in situ rotation. The total plot reward is the accumulation of instantaneous rewards for all steps in a plot. If a collision is detected, this event is immediately terminated with an additional penalty of -10. Otherwise, the plot will continue until it reaches the maximum number of steps (500 steps in our experiment) and ends without penalty.

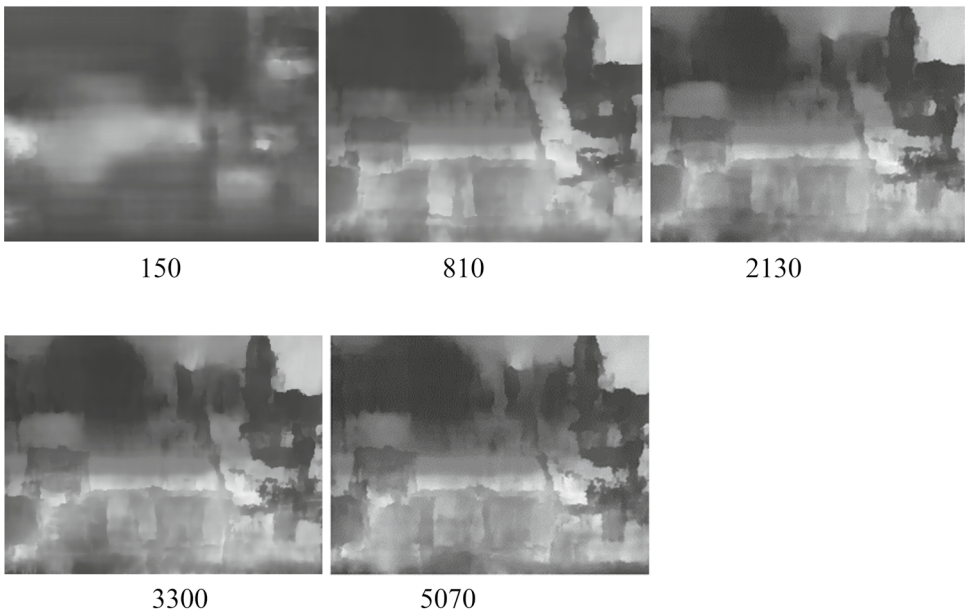

Fig. 8 Depth images of the RGB images were trained 150 times, 810 times, 2130 times, 3300 times and 5070 times

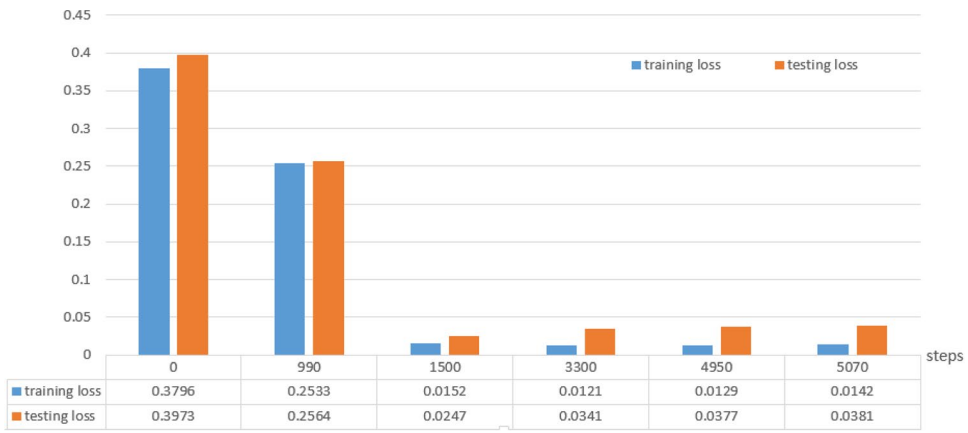

Fig. 9 Loss function of training and testing

(a) Scene images **(b)** RGB-D images **(c)** Depth prediction images

Fig. 10 Depth image and FCRN depth image of the actual scene

5 Simulation and experiment

The simulation environment is built in the Gazebo simulator [12] in Fig. 6. The model before being applied to the real environment needs to be trained in a simulated environment. The presented work uses D3QN model because it is superior to the deep double Q-network (DDQN) and DQN [12]. The model, which is trained in the simulator, is tested directly in several different real-world scenarios.

5.1 Motion prediction of static images

Static image motion prediction: we verify whether the network can predict a reasonable action for arbitrarily complex scenes and avoid obstacles. The training network structure of static image motion prediction is shown in Fig. 7. Figure 8 shows the depth images of RGB images after 150, 810, 2130, 3300 and 5070 training steps, respectively. The loss function obtained from their training is shown in Fig. 9. Note that these scenarios are much more complex than the simulated scenarios used for training, and none of these scenarios have been "seen" by the model before. It can be seen that the trained D3QN model can generate reasonable actions to drive the robot according to the estimated Q value.

5.2 Experiment results

In the real environment, the RGB-D image and the prediction depth image are shown in Fig. 10. The depth image prediction using FCRN network is performed in the real-world environment using the above training result. The actual scene, the depth image and the predicted depth image are shown in Fig. 10. The robot is equipped with a Kinect depth camera to complete the observation function. According to the comparison of the images, the images predicted by the deep neural network can identify the obstacles and predict the next action well, which has a key role in avoiding the obstacles described in Sect. 3.

The obstacles are placed in a room, and the robot can move smoothly in the environment with obstacles. With a laser sensor (Rplidar A2) mounted on top of the robot, the observation function can be completed. The range of input laser distance is the minimum and maximum distance (mm), and the range of angle is set by obtaining the input of the

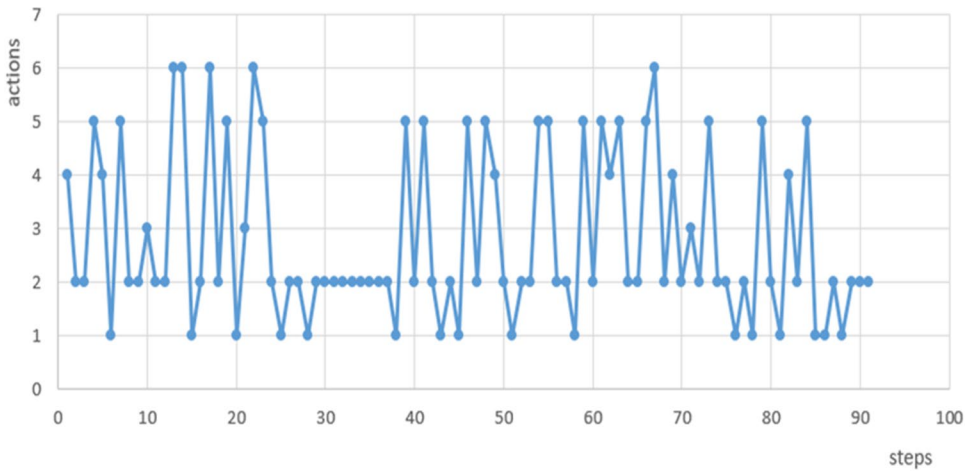

Fig. 11 Action selection of the Turtlebot with two static obstacles

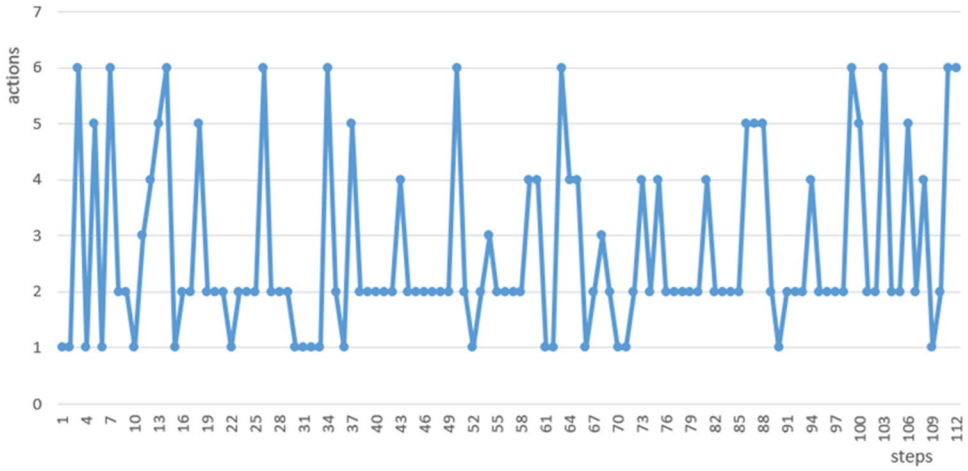

Fig. 12 Action selection of the Turtlebot with four static obstacles

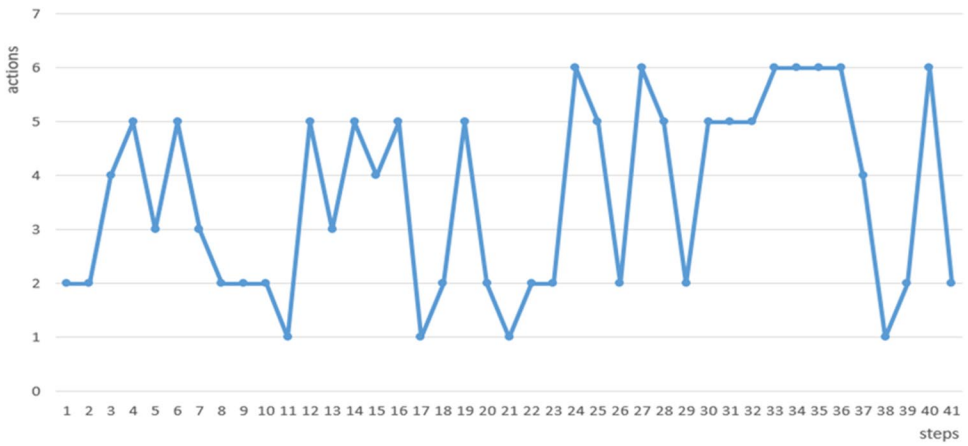

Fig. 13 Action selection of the Turtlebot with two static obstacles and a dynamic obstacle

minimum and maximum angle (radian). The robot drives on a circular road with the minimum curvature to maintain the maximum linear speed. We have performed three sets of experiments: in the environment with (i) two static obstacles, (ii) four static obstacles, and (iii) dynamic obstacles. During the experiment, the actions selection of the Turtlebot is shown in Figs. 11, 12 and 13, respectively. Figure 11 shows the Turtlebot's action selection for two static obstacles, Fig. 12 shows the Turtlebot's action selection for four static obstacles, and Fig. 13 shows the Turtlebot's action selection for two static obstacles and one dynamic obstacle. The choice of action is indicated by 1–6. The action 1 is that the linear velocity is 0.4 m/s, while the angular velocity is $\frac{\pi}{6}$ rad/s. The action 2 is that the linear velocity is 0.2 m/s, while the angular velocity is $\frac{\pi}{6}$ rad/s. The action 3 is that the linear velocity is 0.4 m/s, while the angular velocity is 0 rad/s. The action 4 is that the linear velocity is 0.2 m/s, while the angular velocity is $\frac{\pi}{12}$ rad/s. The action 5 is that the linear velocity is 0.2 m/s, while the angular velocity is $-\frac{\pi}{12}$ rad/s. The action 6 is that the linear velocity is 0.2 m/s, while the angular velocity is $-\frac{\pi}{12}$ rad/s. Figures 10, 11 and 12 demonstrate that the Dueling DQN-based active SLAM robot can avoid static or dynamic obstacles and plan the optimal path.

Figure 14 shows the experimental scenario. In the maps, the robot's walking trajectories are shown by the red curves. The black points are the obstacles, and the black solid circle is the robot. In the trajectory diagrams, the red curves are the walking trajectories of the robot, the black stars are the obstacle, and the yellow circle is the robot. Figure 15 shows the experimental results of two static obstacles in the environment. Figure 15a is the experimental scene, Fig. 15b is the map of the environment with two static obstacles, and Fig. 15c is the path of the avoidance obstacles. The robot can plan an optimal path and finish the map of the environment when there exist two static obstacles. Figure 16 shows the experimental results of four different types of static obstacles in the environment. Three obstacles are cylinder, and one obstacle is square. Figure 16a is the experimental scene, Fig. 16b is the map of the environment with four static obstacles, and Fig. 16c is the path of the avoidance obstacles. The robot can also plan an optimal path and finish

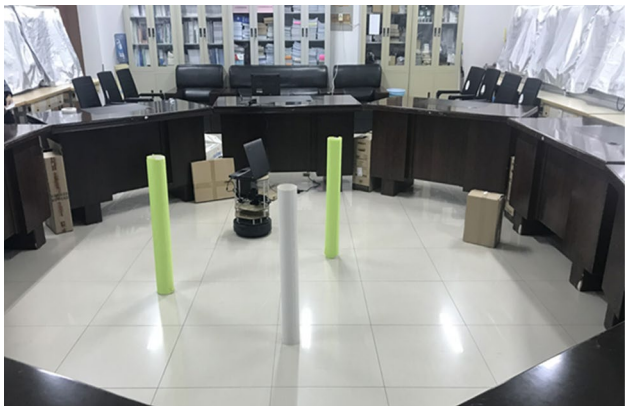

Fig. 14 Experimental scenario

the map of the environment successfully when there exist four different types of static obstacles. Figure 16 shows the experimental results of two static obstacles and one dynamic obstacle in the environment. Figure 17a is the experimental scene, Fig. 17b is the map of the environment with two static obstacles and one dynamic obstacle, and Fig. 17c is the path of the avoidance obstacles. The green line is the optimal path when the robot meets the two static obstacles, and the red line is the new planning trajectory when the robot meets a dynamic obstacle. The blue hollow triangles are the moving route of a dynamic obstacle (the dynamic obstacle is a walking person in this paper). The robot can plan an optimal path and finish the map of the environment when there exist two static obstacles and one dynamic obstacle.

6 Conclusions

This paper proposes path planning based on dueling deep reinforcement learning integrated with FastSLAM under the unknown environment. Dueling deep reinforcement learning algorithm is used for path planning in the unknown environment, while FastSLAM algorithm is used to locate and map the environment. Simulation and experimental results show that this method can successfully carry out autonomous navigation and avoidance path planning in different

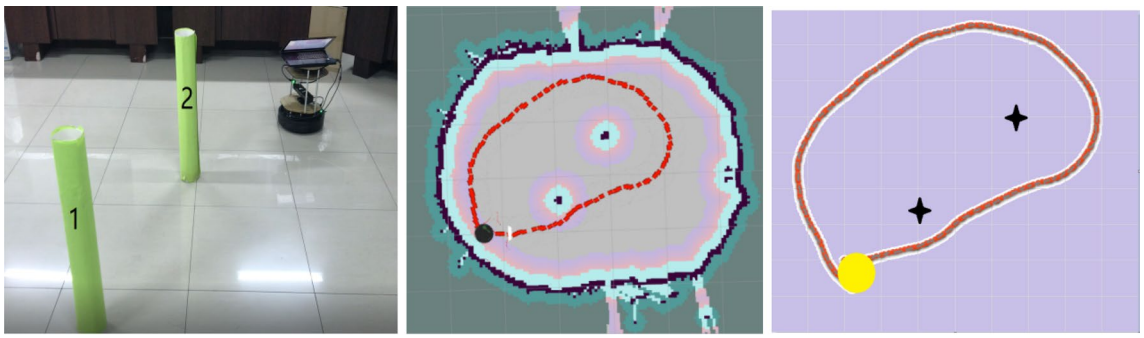

Fig. 15 Corresponding scene (**a**) and the environment map (**b**) and planned trajectory (**c**) with two static obstacles

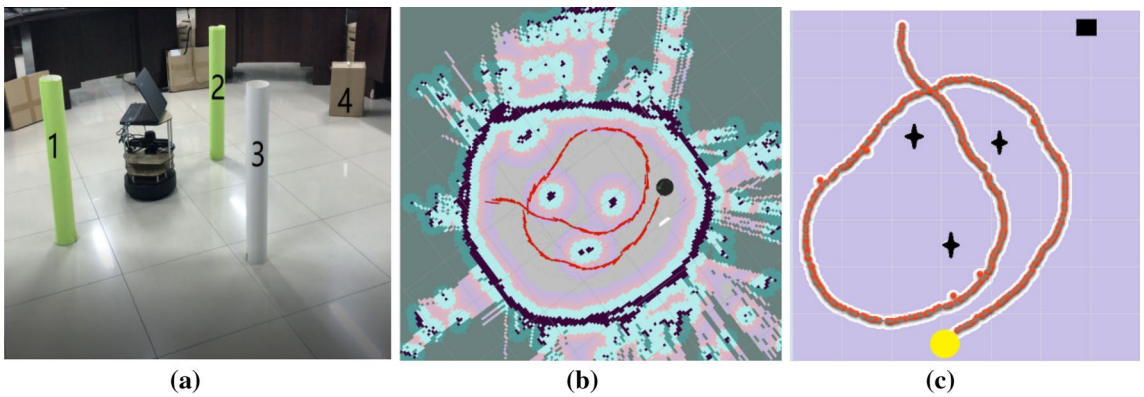

Fig. 16 Corresponding scene(**a**) and the environment map (**b**) and planned trajectory (**c**) with four static obstacles

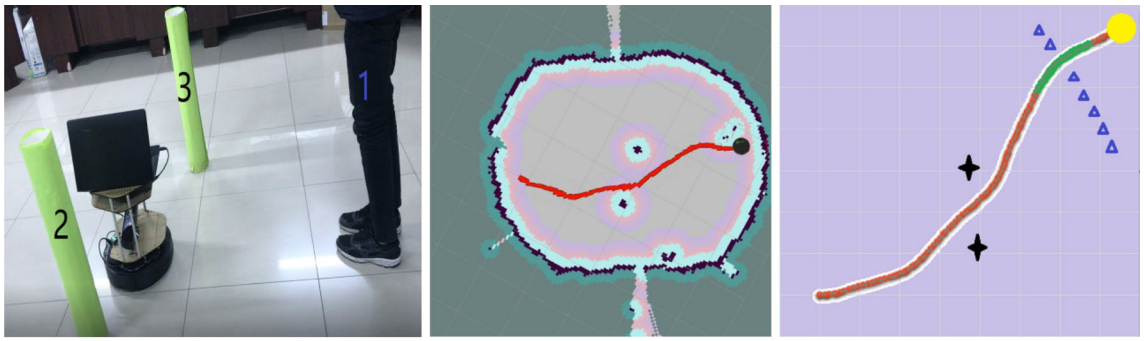

Fig. 17 Corresponding scene (**a**) and the environment map (**b**) and planned trajectory (**c**) with two static obstacles and one dynamic obstacle (a person)

environments. The future work will improve deep reinforcement learning including other sensors and improving the strategy of learning to reduce the time to complete a task.

Acknowledgements This work was supported by the National Natural Science Foundation of China under Grant No. 61773333.

References

1. Chanier F, Checchin P, Blanc C, et al (2008) Map fusion based on a multi-map SLAM framework. In: 2008 IEEE international conference on multisensor fusion and integration for intelligent systems, 2008, MFI. IEEE, pp 533–538
2. Gouaillier D, Collette C, Kilner C (2010) Omni-directional closed-loop walk for NAO. In: 2010 10th IEEE-RAS international conference on humanoid robots (Humanoids). IEEE, pp 448–454
3. Chaves SM, Kim A, Eustice RM (2014) Opportunistic sampling-based planning for active visual SLAM. In: 2014 IEEE/RSJ international conference on intelligent robots and systems (IROS 2014). IEEE, pp 3073–3080
4. Prozorov AV, Priorov AL, Tyukin AL et al (2017) Algorithm for simultaneous localization and mapping based on video signal analysis. Meas Tech 59(10):1088–1093
5. Osswald S, Hornung A, Bennewitz M (2010) Learning reliable and efficient navigation with a humanoid. In: IEEE international conference on robotics and automation. IEEE, pp 2375–2380
6. Wei C, Xu J, Wang C, et al (2013) An approach to navigation for the humanoid robot Nao in domestic environments. In: Conference towards autonomous robotic systems. Springer, Berlin, pp 298–310
7. Fulgenzi C, Ippoliti G, Longhi S (2009) Experimental validation of FastSLAM algorithm integrated with a linear features based map. Mechatronics 19(5):609–616
8. Hornung A, Kai MW, Bennewitz M (2010) Humanoid robot localization in complex indoor environments. In: IEEE/RSJ international conference on intelligent robots and systems. IEEE, pp 1690–1695
9. Berns K, von Puttkamer E (2009) Simultaneous localization and mapping (SLAM). In: Autonomous land vehicles. Springer, Berlin, pp 146–172
10. Havangi R, Taghirad HD, Nekoui MA et al (2014) A square root unscented FastSLAM with improved proposal distribution and resampling. IEEE Trans Ind Electron 61(5):2334–2345
11. Kim DK, Chen T (2015) Deep neural network for real-time autonomous indoor navigation. ArXiv preprint arXiv:1511.04668
12. Giusti A, Guzzi J, Ciresan DC et al (2016) A machine learning approach to visual perception of forest trails for mobile robots. IEEE Robot Autom Lett 1(2):661–667
13. Tai L, Li S, Liu M (2016) A deep-network solution towards model-less obstacle avoidance. In: 2016 IEEE/RSJ international conference on intelligent robots and systems (IROS). IEEE, pp 2759–2764
14. Maček K, Petrović I, Perić N (2002) A reinforcement learning approach to obstacle avoidance of mobile robots. In: 7th international workshop on advanced motion control. pp 462–466
15. Zhou Y, Er MJ (2006) Self-learning in obstacle avoidance of a mobile robot via dynamic self-generated fuzzy *Q*-learning. In: 2006 and international conference on intelligent agents, web technologies and internet commerce, international conference on computational intelligence for modelling, control and automation. IEEE, pp 116–116
16. Laina I, Rupprecht C, Belagiannis V, et al (2016) Deeper depth prediction with fully convolutional residual networks. In: 2016 Fourth international conference on 3D vision (3DV). IEEE, pp 239–248
17. Wen S, Zheng W, Zhu J et al (2012) Elman fuzzy adaptive control for obstacle avoidance of mobile robots using hybrid force/position incorporation. IEEE Trans Syst Man Cybern Part C (Appl Rev) 42(4):603–608
18. Wen S, Chen X, Ma C et al (2015) The *Q*-learning obstacle avoidance algorithm based on EKF-SLAM for NAO autonomous walking under unknown environments. Robot Auton Syst 72:29–36
19. Xie L, Wang S, Markham A, et al (2017) Towards monocular vision based obstacle avoidance through deep reinforcement learning. ArXiv preprint arXiv:1706.09829
20. Wang Z, Schaul T, Hessel M, et al (2015) Dueling network architectures for deep reinforcement learning. ArXiv preprint arXiv:1511.06581
21. Gruslys A, Dabney W, Azar MG et al (2017) The reactor: a fast and sample-efficient actor-critic agent for reinforcement learning. ArXiv preprint arXiv:1704.04651v2
22. Chen J, Bai T, Huang X, et al (2017) Double-task deep *Q*-learning with multiple views. In: Proceedings of the IEEE international conference on computer vision. pp 1050–1058

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.