This is a repository copy of *Obstacle avoidance using stereo vision and deep reinforcement learning in an animal-like robot*.

White Rose Research Online URL for this paper:
http://eprints.whiterose.ac.uk/158352/

Version: Accepted Version

Proceedings Paper:

Ling, F., Jiménez-Rodríguez, A. and Prescott, T.J. orcid.org/0000-0003-4927-5390 (2020) Obstacle avoidance using stereo vision and deep reinforcement learning in an animal-like robot. In: 2019 IEEE International Conference on Robotics and Biomimetics (ROBIO). IEEE International Conference on Robotics and Biomimetics (ROBIO), 06-08 Dec 2019, Dali, China. IEEE , pp. 71-76. ISBN 9781728163222

https://doi.org/10.1109/ROBIO49542.2019.8961639

© 2019 IEEE. Personal use of this material is permitted. Permission from IEEE must be obtained for all other users, including reprinting/ republishing this material for advertising or promotional purposes, creating new collective works for resale or redistribution to servers or lists, or reuse of any copyrighted components of this work in other works. Reproduced in accordance with the publisher's self-archiving policy.

Reuse

Items deposited in White Rose Research Online are protected by copyright, with all rights reserved unless indicated otherwise. They may be downloaded and/or printed for private study, or other acts as permitted by national copyright laws. The publisher or other rights holders may allow further reproduction and re-use of the full text version. This is indicated by the licence information on the White Rose Research Online record for the item.

Takedown

If you consider content in White Rose Research Online to be in breach of UK law, please notify us by emailing eprints@whiterose.ac.uk including the URL of the record and the reason for the withdrawal request.

eprints@whiterose.ac.uk
https://eprints.whiterose.ac.uk/

Obstacle Avoidance Using Stereo Vision and Deep Reinforcement Learning in an Animal-like Robot

Fuhai Ling, Alejandro Jimenez-Rodriguez, and Tony J. Prescott
Department of Electronic and Electrical Engineering, Department of Computer science.
The University of Sheffield
Sheffield, S1 4DP, UK
{fling1& & a.jimenez-rodriguez & t.j.prescott}@sheffield.ac.uk

Abstract—**Obstacle avoidance is a fundamental behavior required to achieve safety and stability in both animals and robots. Many animals perceive and safely navigate their environment using two eyes with overlapping visual fields, allowing the use of stereopsis to compute distances to surfaces and to support collision avoidance. In this paper we develop an obstacle avoidance behavior for the biomimetic robot *MiRo* that combines stereo vision with deep reinforcement learning. We further show that avoidance strategies, learned for a simulated robot and environment, can be effectively transferred to a physical robot.**

Keywords—Reinforcement Learning; Deep Q Network; Stereo Vision; Obstacle Avoidance; MiRo Robot, Animal-like Robot

I. INTRODUCTION

Obstacle avoidance is a fundamental capability required by any mobile robot and is often supported by active sensing systems such as sonar, infrared emitters, or laser range-finders (lidar) [4]. Cameras, on the other hand, are low-cost, light-weight sensors that are available in a wide range of specifications [5], more importantly, they can offer colour and brightness information about the local environment that can support a much richer capacity for scene understanding than range sensing alone.

Whilst the vast majority of animals have eyes, it was previously thought that the capacity for stereo vision was confined to animals with forward-facing eyes such as primates and carnivores , however, stereopsis has now been demonstrated in a wide range of animals including prey mammals with side-facing eyes, birds, amphibians and some invertebrates [19]. Stereo vision is thought to have evolved separately in different vertebrate and invertebrate lineages and to subserve multiple useful functions from range-finding for navigation to camouflage breaking for prey capture.

While some collision avoidance strategies, such as responses to looming stimuli, may be innate, there is a large body of evidence showing the critical role of sensory experience in the development of visually-guided behaviour in mammals [21]. For instance, kittens, whose eyes open around post-natal day ten, frequently bump into objects, achieving adult levels of obstacle avoidance at around postnatal day 26. Experiments with cats reared in the dark for up to 7 months show that a similar period of 14 days visual experience is needed for these animals to achieve normal levels of obstacle avoidance indicating that this milestone specifically requires visual input [22].

The capacity to use bio-inspired reinforcement learning algorithms to acquire obstacle avoidance was demonstrated by Prescott and Mayhew [20] for a simulated robot equipped with an array of three range-finder sensors. This system employed an actor-critic learning algorithm together with coarse-coding function approximators. More recent approaches to learning obstacle avoidance for robots, include research by Xie et al. who trained a convolution neural network for robot depth prediction and obstacle avoidance using a monocular vision system [5].

Deep learning is playing an increasingly importance role in robotics and computer vision where it is being used to support capabilities such as planning path for collision avoidance and navigation robotics [7,8,9,16], however, standard supervised learning approaches typically require a large amount of manually labelled data which has led to the development of deep reinforcement learning algorithms. While these algorithms can learn from sparse and easy to compute reward signals, they suffer the drawback of long training times limiting their applicability for learning in real-world situations. A possible solution is to perform learning in a simulated environment/robot before transferring and fine-tuning the acquired competence on the physical robot platform.

This paper focuses on the problem of obstacle avoidance for a biomimetic robot with non-parallel binocular camera geometry. Specifically, our contributions are:

- To apply stereo imaging for depth estimation using non-parallel robot cameras and to simplify the depth information to provide suitable range information for obstacle avoidance learning.
- To adapt a deep Q-learning network (DQN) [13] to learn obstacle avoidance showing good learning efficiency and performance with limited computational resources.
- To show transfer of learning from a simulated robot to the physical robot platform.

The remainder of this paper is structured as follows. Section II provides a brief introduction to the animal-like MiRo robot, sections III and IV describe our approach to stereo imaging and reinforcement learning respectively. Section IV presents the experiments, results, and discussion.

II. THE MIRO ROBOT

MiRo (Figure 1) is a biomimetic robot platform developed by *Consequential Robotics Ltd* which is a spin-out of the University of Sheffield in the UK. MiRo is a wheeled robot platform equipped with multiple sensor systems and with eight degrees of freedom (DOF) of movement. Rather than specifically emulating any one animal, MiRo is designed to show a number of generic features of mammalian sensorimotor systems, including a 2-DOF neck, and a binocular vision system with non-parallel geometry (illustrated in Figure 1) resembling that of an animal such as a rabbit. In its pre-loaded autonomous mode, MiRo is controlled by brain-inspired control system containing a layered control architecture alongside event-based centralized action selection mechanisms [10]. MiRo provides a useful platform for embodied testing of theories and models of mammalian sensorimotor control, however, since MiRo resembles a companion animal or pet, it is also well suited for applications such as robot-assisted therapy where robots are used as substitutes for animals for therapeutic purposes such as reducing anxiety [6]. MiRo therefore fulfills two roles, serving as a platform for scientific research and as a way of advancing biomimetic robotics towards useful applications.

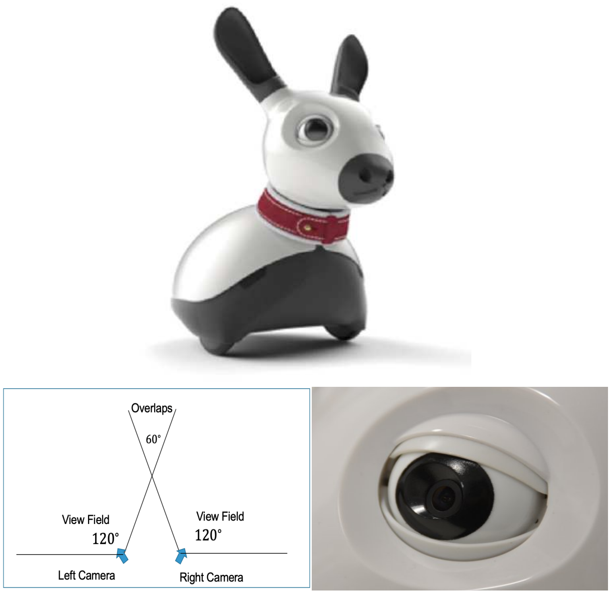

Fig.1: Top. Miro Robot. Bottom. (a) Position of cameras (b) camera on Miro. It has two cameras which are mounted in the eye sockets of an animal-like head. Each camera has a horizontal/vertical field of view of 120/62 degrees and an aspect ratio of 16:9 (pixel aspect ratio is 1:1). The stereo overlap region is a little more than 60 degrees, and the two cameras together provide a wide horizontal field of view of nearly 180 degrees.

III. COMPUTING STEREO FOR THE MIRO ROBOT

Animals with two eyes are able to compute depth by using the disparity between the images obtained by each, specifically, nearby positions in space will have a larger disparity than those that are further away. Similarly, in robotics, we can emulate this capability by identifying corresponding points in the images obtained by two cameras at different positions and angles. Using the baseline distance between two cameras, we can compute the three-dimensional position of the point using the principle of triangulation [19]. The process of depth computation is generally divided into four steps when using two cameras [15,18]:

1) **Undistortion**: there are two main forms of distortion in camera images—radial distortion and tangential distortion. Radial distortion is due to the fish-eye effect that causes rays further from the center of the lens to be bent more than those that closer. Tangential distortion is chiefly caused by the manufacturing issues resulting in the imaging plane not being exactly parallel to plane of the lens. We can use the relationship between the pixel coordinate system and the world coordinate to do the calibration and to calculate the camera parameters including the intrinsic matrix, rotation matrix and translation vector as shown in Fig3. In the calibration process, a known size chessboard plane with 7×9 corners, shown as Fig.2, is applied for capturing images as known position inputs to figure out camera internal and external parameters. We then apply these transforms, as shown in equation 1, to remove tangential and radial lens distortion for each camera.

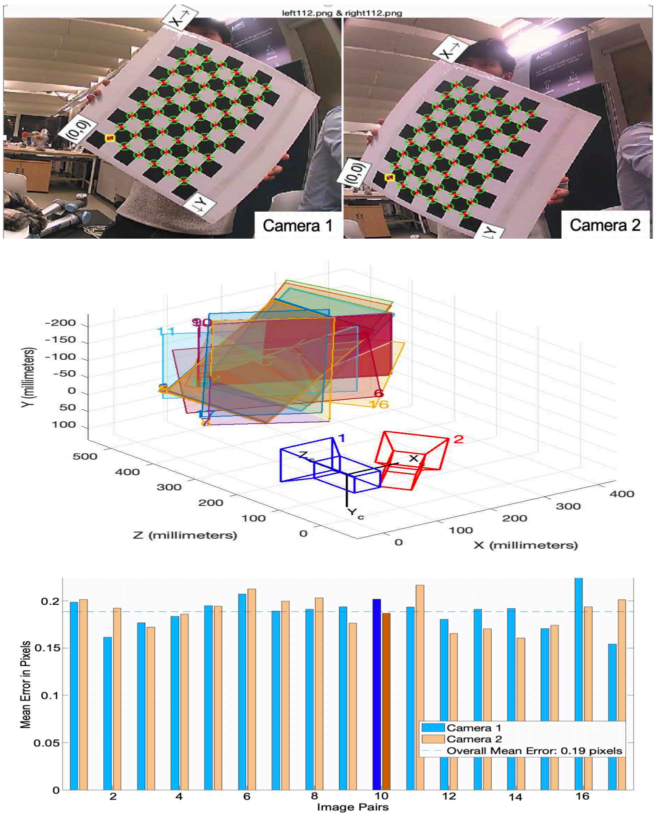

Fig.2: Top. Two camera calibration using a chessboard, the side length of each square is 37.14mm. Middle. Calibration results, using the method described in [15]. with relative position of two cameras using 16 pairs of chessboard images taken at different positions, different angels and different postures. It is illustrated in a three-dimensional coordinate system using the center of left camera image plane as origin. Bottom. Reprojection error after calibration.

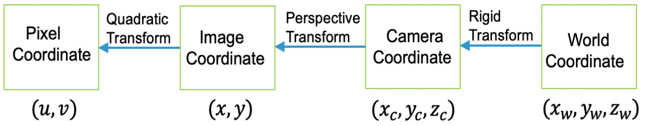

Fig.3:Mathematical transformation

$$z_c \begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = K_{3\times3} \begin{bmatrix} R_{3\times3} & T_{3\times1} \\ 0 & 1 \end{bmatrix} \begin{bmatrix} X_w \\ Y_w \\ Z_w \\ 1 \end{bmatrix} \Rightarrow s \begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = P_{3\times4} \begin{bmatrix} X_w \\ Y_w \\ Z_w \\ 1 \end{bmatrix} \tag{1}$$

Here K_{3X3} is the intrinsic matrix of the camera, P_{3X4} indicates the Perspective projection matrix and *s* is the scale factor.

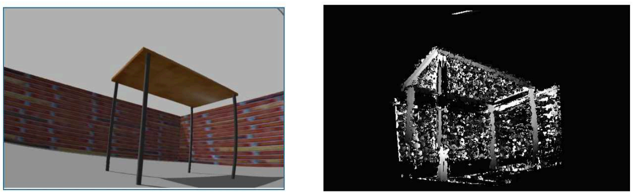

Fig. 4: Disparity maps for the simulated robot.

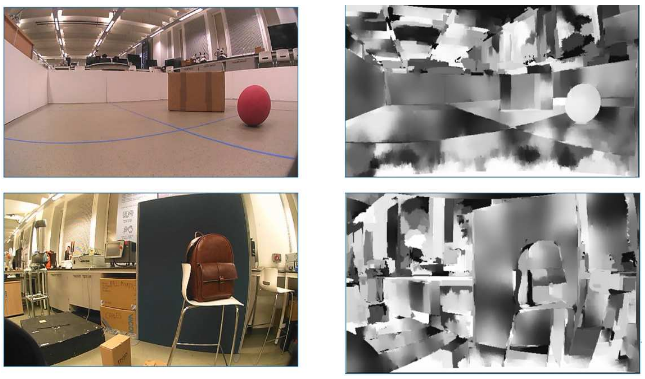

Fig. 5: Disparity maps using the physical MiRo robot.

2) **Rectification**: This step involves adjusting the relative positions between two cameras so as to output images which are row-aligned and rectified using the mathematical transformation between two camera coordinate system from the calibration results(Fig.2). This results in images that are collinear relative to each other.

3) **Correspondence**: This step identifies similar features in the two camera images, computes their disparity (in the x-co-ordinate of the image plane), and outputs the disparity map as shown as figures 4 and 5 for the simulated and physical MiRo robots respectively. There are two main stereo correspondence approach: block matching (BM) and semi-global block matching (SGBM), in this study we use the BM algorithm because it is faster and more reliable in a computational resource limited platform like a mobile robot in our case than SGBM algorithm which requires an order of magnitude more compute time thus be considered not yet suitable for real-time video application[15].

4) **Reprojection**: The final step is to compute the depth map by turning the disparity map into distances using the principle of triangulation and geometric information about the two cameras (as shown in Fig.6). Using the similar triangles, we can derive the depth information as shown in equation 2.

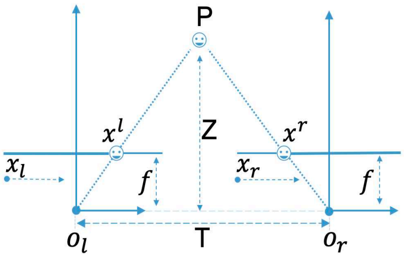

Fig.6: Depth computation using triangulation from two eyes.

$$\frac{T-(x_l-x_r)}{Z-f}=\frac{T}{Z}\Rightarrow \mathrm{Z}=\frac{f\cdot T}{x_l-x_r}\quad . \tag{2}$$

These results demonstrate that it is possible to obtain good depth data from a stereo robot vision system with non-parallel camera geometry.

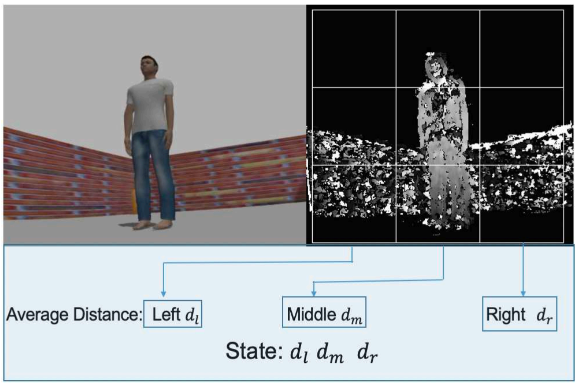

Fig.7: state definition based on depth map

IV. OBSTACLE AVOIDANCE USING STEREO VISION

The stereo vision based obstacle avoidance problem can be considered as a Markov decision process (MDP) where the robot Miro interacts with the environment through a pair of cameras. To be specific, the robot is required to choose an action $a_t \in \mathcal{A}$ depending on the depth information s_t computed from the camera image pair obtained at time $\mathrm{t} \in [0,\mathrm{T}]$. At each step, the robot receives a reward value r_t then transits to the next state s_{t+1}. Hence, the reward function can be denoted as shown in equation 3, where γ is the discount factor, and the aim is to maximize the accumulate future reward G_t.

$$G_t = r_{t+1} + \gamma r_{t+2} + \ldots = \sum_{k=0}^{\infty} \gamma^k r_{t+k+1} \tag{3}$$

Under the policy $a_t = \pi(s_t)$, the Action-value function (Q) is defined as,

$$Q^{\pi}(s,a) = \mathbb{E}[r_{t+1} + \gamma r_{t+2} + \gamma^2 r_{t+3} + \ldots | s,a]$$

which, using the Bellman function, can be rewritten as

$$Q^{\pi}(s,a) = \mathbb{E}[r + \gamma Q^{\pi}(s_{t+1}, a_{t+1})] | s,a] \tag{4}$$

We can derive the optimal Q-value function by choosing the optimal action at each time and state $Q^*(s,a) = \max_{\pi} Q^{\pi}(s,a)$, where

$$Q^*(s,a) = \mathbb{E}_{s+1}[r + \gamma \max_{a_{t+1}} Q^*(s_{t+1}, a_{t+1}) | s,a] \tag{5}$$

It can be seen that the optimal Q-value the robot Miro could earn at current state s_t equals its current reward r_t plus next state's s_{t+1} optimal Q-value multiplied by the discount factor γ. Q-learning uses a lookup table to store and update Q-values for every action at each state, offering simple and efficient approach for optimal Q-value estimation, however, this can result in a large state space which can be costly to store and time-consuming to learn. Traditional approaches have used tiling of the state-space, coarse-coding, or basis functions to reduce the state-space size, alternatively, deep artificial neural networks (ANNs) can be applied to approximate the optimal Q-value function as discussed below. The architecture of the full system, composed of the stereo imaging and two alternative reinforcement learning systems, is shown in Fig. 8.

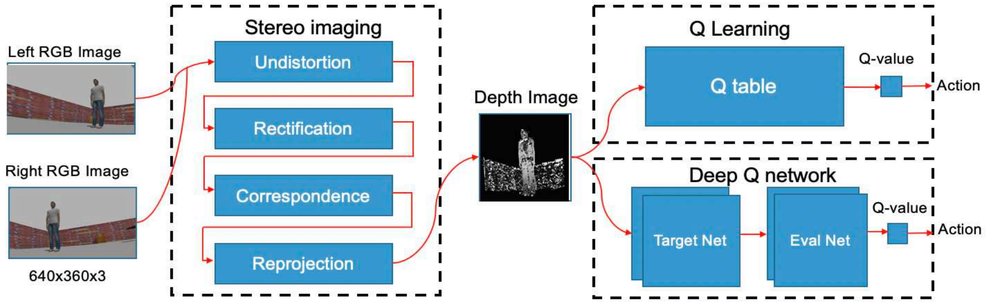

Fig.8 Network architecture of stereo vision based obstacle avoidance using Q-Learning and DQN. A pair of raw RGB images from left and right camera on robot Miro are firstly used to generate depth image by stereo imaging algorithm. And then a Q table is introduced to look up the Q-value of each action and a deep Q network which consists of a target network and an evaluation network to predict the Q-value of left, right and forward action.

A. *Obstacle avoidance using Q-learning*

To evaluate the Q-learning approach, we tile the 640 × 360 disparity map using 3 cells, of 213x360 pixels, and compute the average distance for each cell. However, the average distance $d_t \in [0, \infty]$ in each cell is a continuous number which therefore also needs to be quantised in order to produced discrete cells for the Q-learning algorithm. We therefore set a minimum distance of 140, a maximum of 440, and step-size of 5 (as in equation 6), in order to generate a finite state-space of 60x60x60= 216,000 states:

$$d_t \in [0, \infty] \ \Rightarrow \ d_t \in [140,145,150, \ldots ,440] \qquad (6)$$

[Make this: (d_l, d_m, d_r)= … to match figure 8.]

To learn obstacle avoidance using Q-learning we also need to provide a quantisation of the action space. We define three actions: left or right with an angular speed of 0.79, and forward with an linear speed of 0.1. The reward for each action are shown in table 1. Here we provide a much larger reward value for the forward action in order to encourage the robot to move forward whenever possible. If there is a collision happened the robot receives a negative reward of -10. Additionally, a greedy policy is used in the training with a 0.1 probability of choosing a random action (in order to allow exploration) and 0.9 of choosing the currently optimal action.

Table.1 Reward definition

Action	Forward	Left	Right
Reward	2	0.3	0.3
Punishment (collision)	-10	-10	-10

B. *Deep Q Network*

For continuously-valued state-spaces the Q-learning approach can scale poorly, an alternative method is to combine deep learning with reinforcement learning in order to convert Q-value estimation into a function fitting problem [12]. Specifically, an artificial neural network, that uses the environmental state as input and the Q-values of possible actions as outputs, can be trained to learn an approximation value function as $Q(s, a) \approx f(s, a, \omega)$, where ω denotes the overall parameters of the function f. In order to train the neural network, we need to set up the target Q-value as marked training data based on Q-learning algorithm. In other words, the aim is to build a loss function to calculate the difference between the target Q-value and the evaluation Q-value, which can be calculated as follow,

$$L(\omega) = \mathbb{E}[r + \gamma \max_{a_{t+1}} Q(S_{t+1}, a_{t+1}, \omega) - Q(s_t, a_t, \omega)] \qquad (7)$$

However, combining deep learning and reinforcement learning together also faces some problems:

- First, training a neural network with noisy reward signals can result in a lot of states returning a reward of zero, in other words, the samples are sparse for deep learning.
- Second, the deep learning method requires each sample to be independent of each other, whereas in reinforcement learning the next state is typically strongly dependent on the current state, which means there would be a strong correlation between the training data.

These issues make it difficult for a DQN to converge, however, two methods called experience replay and fixed Q-target network can be applied to resolve these problems [13].

1) **Experience replay**: the experience replay refers to a method that stores the transition samples (s_t, a_t, r_t, s_{t+1}) obtained from the interaction with the environment at each time step in a memory space called the experience pool. This pool is then randomly sampled to train the ANN using gradient decent at each training step. This approach largely resolves the issue of lack of independence between consecutive training patterns.

2) **Fixed Q-target network**: In this approach, a *Q-target network* is constructed that has the same structure as the evaluation network but with different parameters. The Q-target network is used to calculate target Q values instead of the evaluation network and is periodically updated using parameters from the evaluation network. This approach provides a more stable target for training the network. The loss function can be rewritten as

$$I = [r + \gamma \max_{a_{t+1}} Q(S_{t+1}, a_{t+1}, \omega^-) - Q(s_t, a_t, \omega)]^2 \qquad (8)$$

where ω^- denotes the parameters of Q-target network, and ω represents the parameters of evaluation network. The training procedure and corresponding networks are shown in Fig.9. In the experiments described below the DQN has two fully connected layers with 10 artificial neurons in the first hidden layer and 3 in the second hidden layer. The state indicated by Fig.7 is then input into the neural network and corresponding

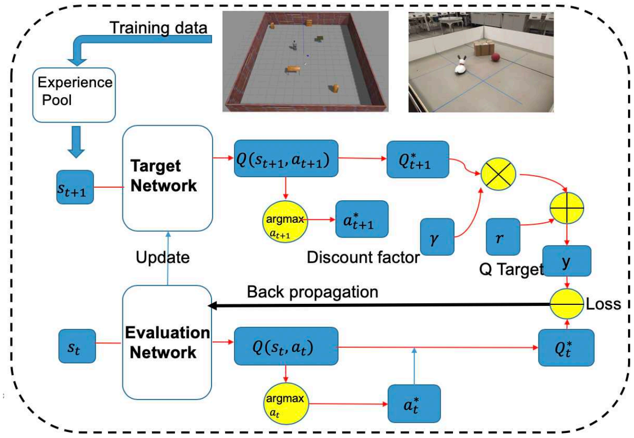

Fig.9: The training procedure of DQN with training data consisting of current state s_t, next state s_{t+1},action a and reward r . $\otimes$,$\ominus$ and $\oplus$ indicate the multiplication, subtraction and addition respectively.

Q-values for left, right and forward action are obtained as outputs, seen as Fig.10. The action and reward specifications were the same as for the Q-learning algorithm.

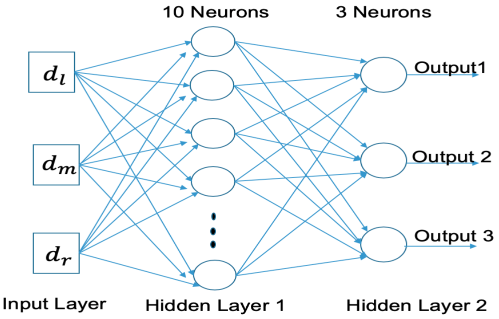

Fig.10: Structure of artificial neural network with two hidden layers used by both target network and evaluation network .

V. RESULTS

The obstacle avoidance system was evaluated both in a simulated environment and in a real indoor environment with the MiRo robot. The simulation environment, built using the Gazebo simulator and shown is Fig.11, consists of a 7×9 m room with normal furniture such as a table and a desk. The simulated MiRo robot is first trained in this room before the actual robot is further trained in an indoor laboratory environment. For each episode, the total reward is calculated as the sum of rewards over all time-steps. Each episode runs for a 1000 steps unless a collision is detected in which case a reward of -10 is delivered followed by immediate termination of the episode.

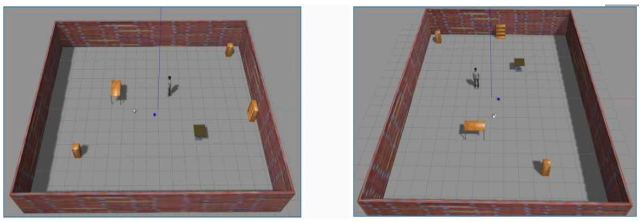

Fig.11: Simulation environments

A. *Training efficiency of alternative learning methods*

In order to compare the performance and efficiency of the two learning methods the results for Q-learning and DQN are shown in Fig. 12.

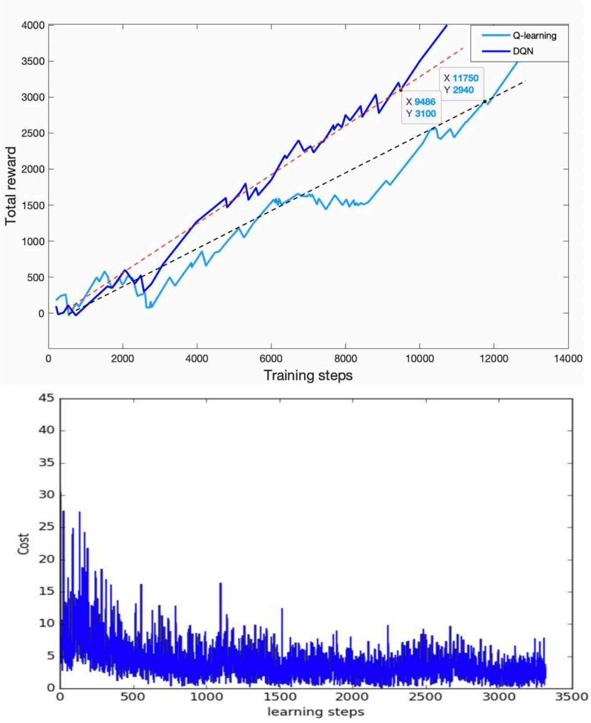

Fig.12 Top. Training curve for Q-learning and DQN. Bottom. Cost curve of DQN.

Each graph shows plots total reward against cumulative steps since the beginning of training. Each system was trained for 20 hours. There is a clear overall trend for total reward to increase as the number of training steps rise. To compare the average reward earned by both algorithms, a dotted straight line that follows the trend of total reward vs training steps and passes through the origin is shown. We can see that the average reward per step received by DQN is approximately 3100/9486=0.327, which is much larger than the one by Q-learning 2940/11750=0.25. Overall, the learning curve is steeper for DQN and reaches a much higher overall level of reward.

Fig.12.Bottom. illustrates the cost curve between target network and the evaluation network with the increasing learning steps for the DQN method. The reduction in cost over time shows that the Q-values estimated by the evaluation network increasingly approach those of the target network demonstrating that the network is converging.

B. *Obstacle avoidance Test.*

After training the robot in the simulation environment and real world, experiments were conducted to test the model. Simulated rooms built in the Gazebo and a 3x3m square space with obstacles were used to test the robot, while the route followed and any collisions that occurred were recorded (for the real robot this was done using an overhead camera). As shown in Fig.13, we can see that it shows a perfect obstacle avoidance behavior both in the simulation and real environment. In each case, the robot runs around the room and avoid collisions with the wall, or with objects, while exploring the room with primarily forward movement.

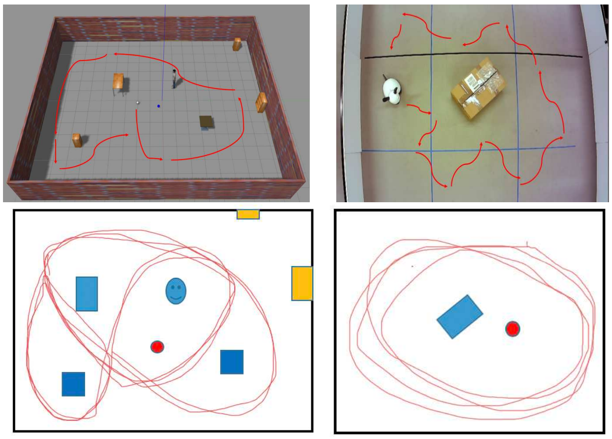

Fig.13: robot running routes in real (right column) and simulated (left column) environments using the DQN network.

Conclusion

This paper has investigated the use of Q-learning and deep reinforcement learning algorithms to allow obstacle avoidance behavior for a biomimetic robot using depth data computed from a stereo camera array. The model has been trained in the simulation and tested in the real environment. DQN, as an method using artificial neural network to predicted the Q-value, offers higher learning efficiency than normal Q-learning algorithm. Experiments and tests both in the simulator and real world show that the robot formed an effective obstacle avoidance behaviour while appropriately exploring the space.

ACKNOWLEDGEMENT

This work was supported by the EU H2020 Programme as part of the Human Brain Project (HBP-SGA2, 785907).

REFERENCES

[1] Tony J. Prescott, Ben Mitchinson, and Sebastian Conran “MiRo- An Animal-like Companion Robot with a Biomimetic Brain-based Control System”, ACM/IEEE International Conference on Human-Robot Interaction, Pages 50-51, march 2017

[2] D. Feil-Seifer and M. J. Matari´c, “Defining socially assistive robotics,” in Proc. IEEE Int. Conf. Rehabilitation Robotics (ICORR’05), Chicago, IL pp. 465–468,June 2005,.

[3] Hemminghaus, J.; Kopp, S. Towards Adaptive Social Behavior Generation for Assistive Robots Using Reinforcement Learning. In Proceedings of the 2017 ACM/IEEE International Conference on Human-Robot Interaction, Vienna, Austria; ACM: New York, NY, USA, 2017; pp. 332–340, 6–9 March 2017

[4] CDR HR Everett. Survey of collision avoidance and ranging sensors for mobile robots. RAS , 5(1):5–67, 1989

[5] L. Xie, S. Wang, A. Markham, and N. Trigoni, “Towards monocular vision based obstacle avoidance through deep reinforcement learning,” ,arXiv preprint arXiv:1706.09829, 2017.

[6] Panagiotidi M, Wilson S and Prescott T (2019) Exploring the Potential of the Animal-Like Robot MiRo as a Therapeutic Tool for Children Diagnosed with Autism. In: Martinez-Hernandez U, Vouloutsi V, Mura A, et al. (eds) *Biomimetic and Biohybrid Systems.* Cham: Springer International Publishing, 351-354.

[7] Ronald Clark, Sen Wang, Niki Trigoni Andrew Markham, and Hongkai Wen. Vidloc: A deep spatio-temporal model for 6-dof video-clip relocalization. In CVPR , 2017.

[8] Ronald Clark, Sen Wang, Hongkai Wen, Andrew Markham, and Niki Trigoni. Vinet: Visual-inertial odomety as a sequence-to-sequence learning problem. In AAAI, pages 3995–4001, 2017.

[9] Alessandro Giusti, J´erˆome Guzzi, Dan C Cires¸an, Fang Lin He, Juan P Rodr´ıguez, Flavio Fontana, Matthias Faessler, Christian Forster, J¨urgen Schmidhuber, Gianni Di Caro, et al. A machine learning approach to visual perception of forest trails for mobile robots. RA Letters , 1(2):661–667, 2016.

[10] Tony J. Prescott, Ben Mitchinson, “MiRo: Social Interaction and Cognition in an Animal-like Companion Robot”, ACM/IEEE International Conference on Human-Robot Interaction, Pages 41-41, march 2018.

[11] Tony J. Prescott, Ben Mitchinson, and Sebastian Conran “MiRo- An Animal-like Companion Robot with a Biomimetic Brain-based Control System”, ACM/IEEE International Conference on Human-Robot Interaction, Pages 50-51, march 2017.

[12] Mnih, Volodymyr, Kavukcuoglu, Koray, Silver, David, Graves, Alex, Antonoglou, Ioannis, Wierstra, Daan, and Riedmiller, Martin. Playing atari with deep reinforcement learning In NIPS Deep Learning Workshop,2013,.

[13] Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Andrei A Rusu, Joel Veness, Marc G Bellemare, and Alex et. al Graves. Human-level control through deep reinforcement learning. Nature , 518(7540):529–533, 2015.

[14] Charis Watkins, C. J. C. H. & Dayan, P. Q-learning.,Mach. Learn. 8, 279–292,1992.

[15] G. Bradski and A. Kaehler, Learning OpenCV: Comput. Vision with the OpenCV Library. Sebastopol, CA, USA: O’Reilly Media, 2008.

[16] Sen Wang, Ronald Clark, Hongkai Wen, Niki Trigoni, R Clark, S Wang, H Wen, N Trigoni, A Markham, A Markham, et al. Deepvo: towards end-to-end visual odometry with deep recurrent convolutional neural networks In ICRA , 2017.

[17] T. Mitchell, Machine Learning. McGraw-Hill, inter. ed., 1997.

[18] Zhang, Z. A flexible new technique for camera calibration. IEEE TPAMI, 22(11):1330–1334. 2000

[19] Nityananda V and Read JCA (2017) Stereopsis in animals: evolution, function and mechanisms. *The Journal of Experimental Biology* 220(14): 2502.

[20] Prescott TJ and Mayhew JEW (1992) Obstacle avoidance through reinforcement learning. In: Moody JE, Hanson SJ and Lippmann RP (eds) *Advances in Neural Information Processing Systems 4.* Denver: Morgan Kaufman.

[21] Tees RC (1986) 11 - Experience and Visual Development: Behavioral Evidence. In: Greenough WT and Juraska JM (eds) *Developmental Neuropsychobiology*. Academic Press, pp.317-361.

[22] Van Hof-Van Duin J (1976) Development of visuomotor behavior in normal and dark-reared cats. *Brain Research* 104(2): 233-241