Preprints of the 21st IFAC World Congress (Virtual)
Berlin, Germany, July 12-17, 2020

Autonomous Reinforcement Control of Underwater Vehicles based on Monocular Depth Vision ⋆

Pengli Zhu * **Shuhan Yao** * **Yancheng Liu** * **Siyuan Liu** ** **Xiaoling Liang** *,***

* *Marine Engineering College, Dalian Maritime University, Dalian, China, (e-mail: dlmu.p.l.zhu@gmail.com).*
** *Department of Radiology and Biomedical Research Imaging Center (BRIC), University of North Carolina at Chapel Hill, Chapel Hill, USA, (e-mail: dmu.s.y.liu@gmail.com).*
*** *Department of Electrical and Computer Engineering, National University of Singapore, Singapore, Singapore, (e-mail: liangxl@dlmu.edu.cn).*

Abstract: In this paper, a monocular depth prediction based end-to-end reinforcement control framework is proposed for autonomous control of underwater vehicles in the unknown environment. In the control framework, with the input of camera sensor RGB videos, a monocular depth prediction network is proposed to generate underwater depth images and a sequential reinforcement learning controller is also developed for autonomous obstacle-avoiding navigation and movement control. Simulated and experimental results demonstrate that the proposed control scheme can achieve remarkable performance on collision-avoidance navigation and autonomous control in the unknown environment.

Keywords: Monocular depth prediction; Autonomous reinforcement control; Underwater vehicles

1. INTRODUCTION

Underwater vehicles recently have been significant tools for marine science with the advantages of low cost, small size, lightweight, high flexibility and wide range of activities. The precise movement control of underwater vehicles plays a decisive role in the safe accomplishment of some high-risk tasks, such as undersea oil exploration, undersea search and rescue and submarine pipeline repair, etc. However, underwater vehicles are highly susceptible to the complex ocean current and fluid resistance, especially in the unknown environment, the environmental changes are unpredictable. Thus, it is highly desired to design a collision-avoidance navigation based autonomous control scheme for the safety assurance of underwater vehicles during tasks.

To address this issue, many control methods are proposed, which mainly fall into two groups: navigation based tracking control schemes(Shen et al. (2016), He and Zhou (2010) and Repoulias and Papadopoulos (2005)), which sequentially perceive environment and design obstacle-avoiding navigation laws and trajectory tracking controllers based on control theories, and end-to-end intelligent control schemes(Carlucho et al. (2018)), which design the controllers using intelligent algorithms (e.g., neural networks, fuzzy logical systems, etc.) by integrating a decision-making based navigation design. The navigation based tracking control schemes aim to search an available path or trajectory from perceptual environmental information and then develop a controller to track the searched path or trajectory accurately. However, these methods encounter some limitations, such as cumbersome control links, complex environmental perception, and imprecise system model. In contrast, the end-to-end intelligent control schemes aim to search optimal action of each sample time from the obtained environmental information so as to behave like human beings. Particularly, deep reinforcement learning (DRL) that inherits both the feature extraction capability of deep learning and the decision-making mechanism of reinforcement learning demonstrates high potential to autonomous control in recent years.

In this paper, a monocular depth vision based reinforcement control framework is proposed for the autonomous control of underwater vehicles subject. With the environmental video as input, a geometric network (GeoNet) based on encoder-decoder framework is proposed to generate depth maps that provides the spatial geometric information of the actual complex environment. Sequentially, a reinforcement learning control network (CtrlNet) built based on the convolutional neural network and double Q learning techniques outputs action decisions for obstacle-avoiding based autonomous control. The effectiveness and

⋆ This work is supported by National Natural Science Foundation (NNSF) of China under Grant 51979021, 51479018 and 51709028, Natural Science Foundation of Liaoning province under Grant 20170520430, and fundamental research funds for the central universities under Grant 3132019317 and 3132018253.
Corresponding author: Yancheng Liu, Siyuan Liu.

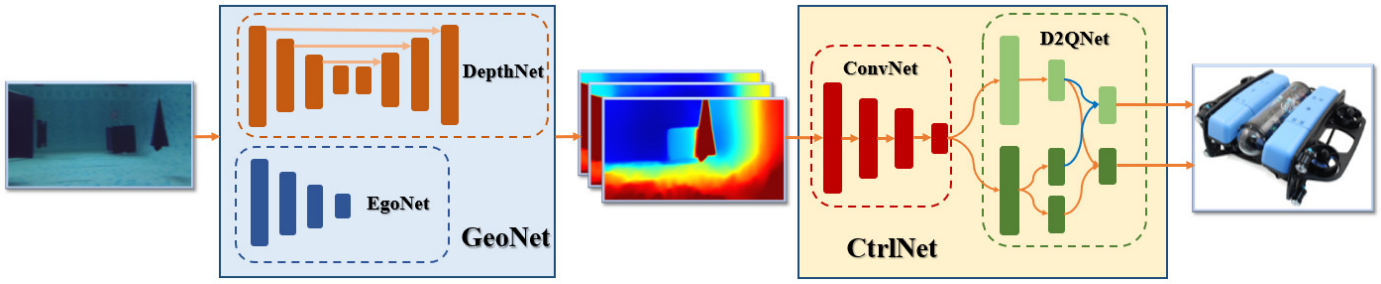


Fig. 1. The overall frame diagram of this paper(The first part is the visual depth prediction part and the latter part is the deep reinforcement learning control part)

superiority of the proposed control scheme are verified by both simulation and experiment studies.

The rest of this paper is organized as follows. The proposed GeoNet and CtrlNet are described in Section 2 and Section 3, respectively. Simulation and experiment studies are given in Section 4, followed by conclusions drawn in Section 5.

2. FROM APPEARANCE TO GEOMETRY

To apply DRL for the motion control of underwater vehicles in practice, a feasible solution is to train models in a simulator and then transfer the trained model to real world application. However, it is highly challenging for vision based control schemes due to the significant visual differences between virtual and real underwater environments. Some researchers use stochastic noise to capture the natural conditions of the real environment (Li and Snavely (2018)). The distance information of surrounding objects is significant for obstacle avoiding and motion state adjustment of underwater vehicles is obtained by the depth and ego-motion estimation of real underwater environments. To this end, the GeoNet is proposed to extract spatial geometric representations (i.e., depth maps and ego-motion estimation) from two consecutive RGB frames of the underwater environment.

2.1 Network Architecture

The GeoNet consists two parts: depth prediction and ego-motion estimation. For depth prediction, an encoder-decoder architecture with residual blocks (He et al. (2016)) is adopted to generate depth maps from RGB frames. For ego-motion estimation, as illustrated in Fig. 2, an ego-motion estimation network takes two RGB frames as input to generate the position transformation matrix between two frames, and regulate the translation and rotation parameters between two frames. As different perspectives of the scene can be predicted by converting one frame into an adjacent frame, the ego-motion estimation for the next frame can be realized by mapping current frame to the next frame. Specifically, when two frames of RGB image I_i and I_j are input, the estimate of ego-motion $E_{i\to j}$ from I_i and I_j can be predicted using PoseNet, which contains seven feature extraction layers of convolution. The depth mapping D_j can be obtained by inputting I_j to the depth prediction network. With the depth map D_j and the estimate of ego-motion $E_{i\to j}$, frame I_i can be warped to $\hat{I}_{i\to j}$ via a translation ϕ (Casser et al. (2019)).

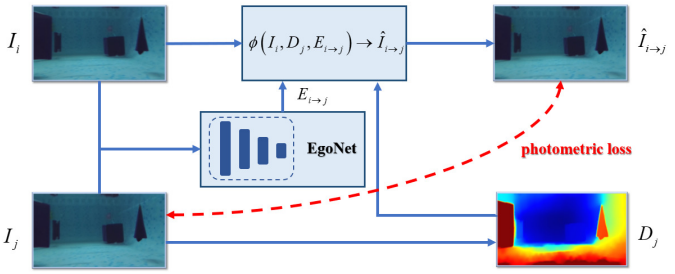


Fig. 2. The overall block diagram of ego-motion estimation network

2.2 Loss Functions

The warping of $\hat{I}_{i\to j}$ performs a translation from I_i to I_j, thus theoretically, the warped frame $\hat{I}_{i\to j}$ and frame I_j should be consistent. To this end, a multi-scale consistency (MSC) loss on the scale of pixel and SSIM between $\hat{I}_{i\to j}$ and I_j is proposed as follows:

$$L_{MSC}^{(j)} = \alpha_1 \min\left(\|\hat{I}_{i\to j} - I_j\|\right) + \alpha_2 L_{SSIM}^{(j)}(\hat{I}_{i\to j}, I_j) \quad (1)$$

where α_1 and α_2 are hyperparameters used for the trade-off between pixel and structural information. Note that the pixel-wise consistency loss is computed as the minimum per-pixel photometric loss to avoid being penalized due to out-of-view pixel and occlusion effects. In addition to MSC loss, a depth smoothness loss is also used to regularize the depth estimates(Zhou et al. (2017)), thus the total loss function of GeoNet is given by:

$$L_{Geo} = \sum_{j=0}^{2}\left(L_{MSC}^{(j)} + \alpha_3 \frac{1}{2^i} L_{SM}^{(j)}\right) \quad (2)$$

where α_3 is a hyperparameter. The depth smoothness loss L_{SM} is constructed using the image gradients, which inflect the sharpness changes of depth at pixel coordinates (Mahjourian et al. (2018)), as follows:

$$L_{SM}^{(j)} = \sum_j \|\partial_x D^j\| e^{-\|\partial_x I^j\|} + \|\partial_y D^j\| e^{-\|\partial_y I^j\|} \quad (3)$$

where ∂_x and ∂_y, respectively, take the gradients on x and y axis of an image, I^j and D^j, are the raw RGB frames and the corresponding depth maps.

2.3 Training Settings: GeoNet

The visual sensor built in underwater vehicles is used to collect real underwater videos, extract and establish

the dataset for training depth prediction network. The network topology is an hour-glass structure in which ResNet18 is utilized as the encoder. During training, we set hyperparameters $\alpha_1 = 0.8$, $\alpha_2 = 0.1$, $\alpha_3 = 0.1$, and we used the Adam optimizer with $\rho = 0.0001$ as the initial learning rate. After training and testing spatial geometric representations network and generate a network model with high quality.

3. FROM GEOMETRY TO POLICY DECISION

Since deep Q network (DQN)(Mnih et al. (2015)) has been proved to be trainable directly benefit from raw images, most DQN models used for automatic control are based on this version (Tai and Liu (2016); Zhang et al. (2016)). Although this architecture ultimately achieves reasonable results, it tends to overestimate Q-value and require additional computational cost for training in the simulator. To this end, a double Q-learning architecture(Wang et al. (2015)) is inherent in CtrlNet to improve the performance on training efficiency of autonomous control based monocular vision.

3.1 Problem Formulation

The autonomous control based on monocular vision problem can be regarded as the "perception-decision" process in which the underwater vehicle interact with the environment through monocular camera. The underwater vehicle chooses an action $a_t \in A$ according to the depth frame x_t at time $t \in [0, T]$, then observe a reward signal r_t produced by a reward function and transit to the next observation x_{t+1}. The main objective of this algorithm is to maximize the accumulative future reward $R_t = \sum_{\tau=t}^{T} \gamma^{\tau-t} r_\tau$, where γ is the discount factor. The direction and growth rate of the reward value determine the training speed and performance.

Given the policy $\pi(\cdot)$ used to generate actions and yields obtaining action $a_t = \pi(x_t)$ with the current state x_t in one iteration, the Q-value of a state-action pair (x_t, a_t) can be defined as follows

$$Q^{\pi}(x_t, a_t) = \mathrm{E}[R_t | x_t, a_t, \pi]. \tag{4}$$

For convenient implementation, the above formula is computed by using the Bellman equation(Bellman and Kalaba (1965))

$$Q^{\pi}(x_t, a_t) = \mathrm{E}\left[r_t + \gamma \mathrm{E}\left[Q^{\pi}(x_{t+1}, a_{t+1}) | x_t, a_t, \pi\right]\right]. \tag{5}$$

By choosing the optimal action during each iteration where $Q^*(x_t, a_t) = \max_{\pi} \mathrm{E}[R_t | x_t, a_t, \pi]$, the optimal Q-value can be obtained by

$$Q^*(x_t, a_t) = \mathrm{E}_{x_{t+1}}\left[r + \gamma \max_{a_{t+1}} Q^*(x_{t+1}, a_{t+1}) | x_t, a_t\right] \tag{6}$$

which indicates that the optimal Q-value can be obtained at time t is the summation of the current reward r_t and the discounted optimal Q-value available at time $t+1$. revRather than calculating the Q-value function directly over a large state space, the problem can be solved by using a deep neural network to approximate this optimal Q-value function.

3.2 Network Architecture

As illustrated in Fig.1, CtrlNet, the latter part of the overall frame diagram, consists two parts: (1) convolution network (ConvNet, red in Fig. 1), which is constructed a four-layer fully-convolutional neural network to extract features from the input depth maps; and (2) deep double Q-network structure(D2QNet, green in Fig.1), which is constructed two fully-connected laminar flows (i.e., a state value function (SVF) and an action advantage function (AAF)) to estimate states and select actions, respectively. With the estimated states and selected actions, the Q-value function is constructed as

$$Q(x, a; \theta, \alpha, \beta) = V(x; \theta, \beta) + A(x, a; \theta, \alpha) \tag{7}$$

where V and A, respectively, represent SVF and AAF. Note that the AAF indicates the difference between current performance and average performance. That is, if the advantage value is larger than the average, then the value of AAF is positive and vice versa. Assuming that the expectation of AAF is zero, then the Q-value function can be rewritten as

$$\begin{aligned} Q(x, a; \theta, \alpha, \beta) =& V(x; \theta, \beta) + \Big(A(x, a; \theta, \alpha) \\ & - \frac{1}{|\mathcal{A}|} \sum_{a'} A(x, a'; \theta, \alpha)\Big) \end{aligned} \tag{8}$$

where $|\mathcal{A}|$ is the cardinal number of the AAF, which is equal to the size of the action set. If each A value is subtracted from the average of all A values during the iteration, the zero-expectation constraint is achieved and yields stability enhancement of the overall output.

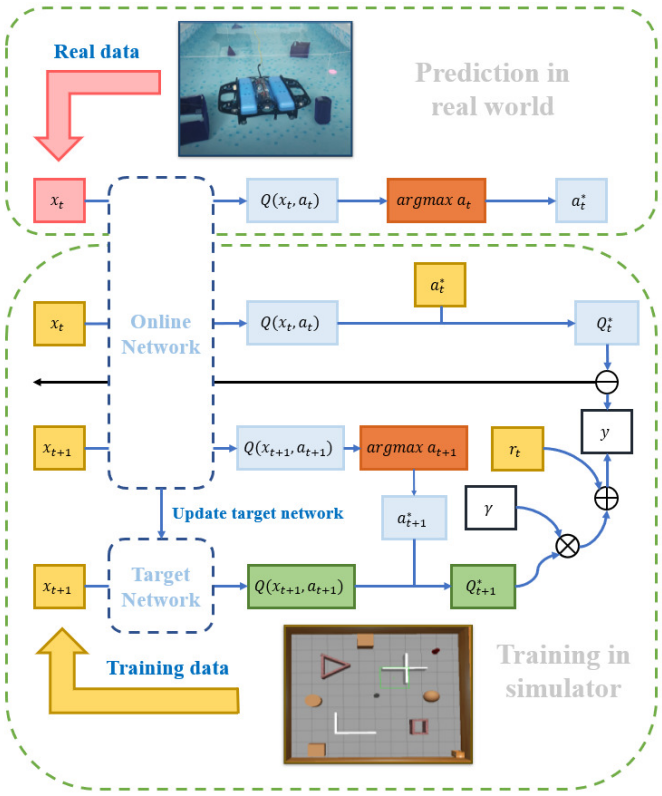


Fig. 3. The training procedure of D2QNet. $\oplus, \ominus, \otimes$ are element-wise addition, subtraction and multiplication, respectively.

As illustrated in Fig. 3, D2QNet utilizes a target network alongside an online network. The target network is a duplicate of the online one. revUnlike the online network that updates the weights through back propagation at each training step, the weights of the target network are fixed for a short period and then copied from the online network. Based on the dual-network setting, the online and target networks are designed for action selection and optimal state estimation, respectively. Specifically, with the input state x_{t+1}, the online and target networks are used, respectively, to select optimal action a^*_{t+1} from Eq. (6) and to calculate optimal Q value Q^*_{t+1} at time $t+1$. Then, with the discount factor γ and current reward r_t, the target value y at t can be obtained as follows:

$$y = \begin{cases} r & \text{if } x' \text{ is terminal} \\ r + \gamma Q^* \left(x_{t+1}, a^{\max}\left(x_{t+1}; \theta\right); \theta^-\right) & \text{otherwise.} \end{cases} \tag{9}$$

where $a^{\max}\left(x_{t+1}; \theta\right) = \arg\max_{a'} Q\left(x_{t+1}, a'; \theta\right)$, x_{t+1} is the next observation at time $t+1$, θ and θ^- are the parameters of online network and target network, respectively.

Finally, the loss value is calculated by subtracting the target value with the optimal value Q^* predicted by the online network, and then back-propagated to update the weights of online network with a gradient descent step, where the CtrlNet loss function L_{Ctrl} is designed as

$$L_{Ctrl} = \|y - Q(x, a; \theta)\|^2 \tag{10}$$

3.3 Training Settings: CtrlNet

From the practical requirements of autonomous control of the underwater vehicle, 7 actions including two settings of linear velocity (0.4 or 0.6 m/s) and five settings of angular velocity (i.e., $\frac{\pi}{3}$, $\frac{\pi}{6}$, 0, $-\frac{\pi}{6}$, $-\frac{\pi}{3}$ rad/s) are integrated into an action set. The instantaneous reward function is defined as

$$r = v \cdot cos\alpha \cdot \Delta T \tag{11}$$

where v is linear velocity, α is angular velocity, and ΔT is the time for each episode. The total episode reward is the accumulation of instant rewards for all steps in an episode. If a collision is detected, the episode ends immediately with a additional penalty of -5. Otherwise, the episode continues until the maximum number of steps is reached and ends without penalty. The learning rate is set to 10^{-5} in an Adam optimizer(Kingma and Ba (2014)), the discount factor $\gamma = 0.99$, the capacity of replay memory buffer for storing state x_t, action a_t and reward r_t information is 50000 and the parameter update rate from target network to online network is 0.001.

4. SIMULATION AND EXPERIMENT STUDIES

In this section, the effectiveness of the proposed autonomous control algorithm of underwater vehicles is evaluated by both simulation and experiment studies.

4.1 Dataset

To satisfy the experimental requirement, experiments are conducted on in-house underwater scene dataset containing various shaped obstacles. The resolution of each frame is 1920 × 1080 and corresponding rate is 15fps. The in-house dataset contains total 2780 images (2228 training images and 552 testing images). The input patches of GeoNet are cropped with a uniform size 416 × 128. The size of the input depth image for CtrlNet is also 416 × 128. The training data of CtrlNet is generated by its own exploration in the simulator.

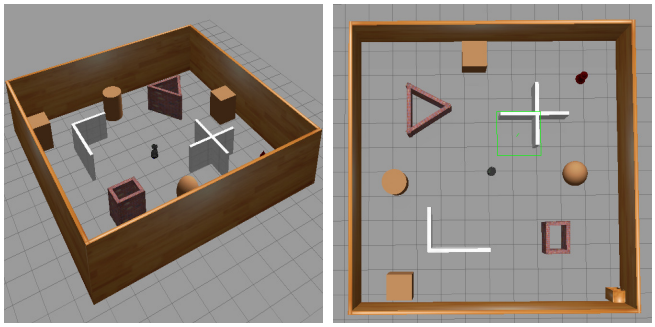


Fig. 4. Robot training simulator-Gazebo

4.2 Implement Details

The proposed method is implemented using Tensorflow (Abadi et al. (2016)). During simulation and experiment, the underwater vehicle gets continuous raw RGB frames from the monocular camera, which are then inputted into GeoNet to generate depth maps for CtrlNet. The CtrlNet extracts spatial geometric information from depth maps and then aggregates them to output linear velocity and angular velocity for the underwater vehicle movement. It is worth noting that the output linear and angular velocities are the probability values of the actions in action sets defined above. The above is the overall integrated closed-loop system. By asynchronously integrating GeoNet and CtrlNet, robustness operation of the overall system is achieved.

Simulation To verify the effectiveness of visual depth prediction algorithm and collision-avoiding based reinforcement control algorithm for the underwater vehicle during training, the proposed model is trained in Gazebo simulator with ROS (Robot Operating System) environment using two GPUs (i.e., NVIDIA GeForce GTX 2080Ti 11GB). During the simulation training, the underwater vehicle interacted with an external computer using ROS. The simulation environments with many obstacles are built in Gazebo simulator as shown in Fig. 4.

As shown in Fig. 5, the reward value increases rapidly with the number of iteration, reaching a relatively stable reward value after 900 iterations. The path trajectory of the simulation underwater vehicle in Gazebo as shown in Fig. 6, from which it is observed that the underwater vehicle usually chooses to a similar path for obstacle avoiding. revThis is because after obtaining the Q value of each state, the behavior will be predicted by the network and selected by a greedy policy, such that resulting in a fixed policy for all states. Since the reward function defined during the training phase tends to keep straight rather than turning, the underwater vehicle navigates as a loop with minimal curvature to maintain a maximum linear speed and successfully avoid all collisions. This indicates

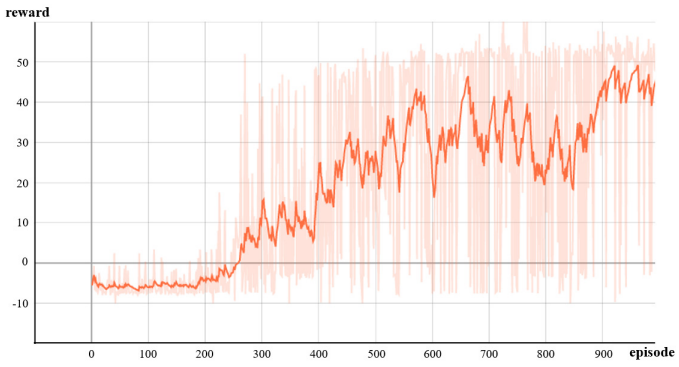


Fig. 5. The reward curve with a smoothness of 0.9 in the simulator

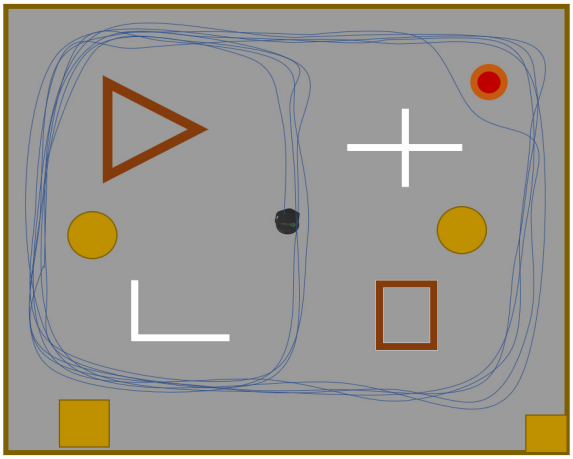
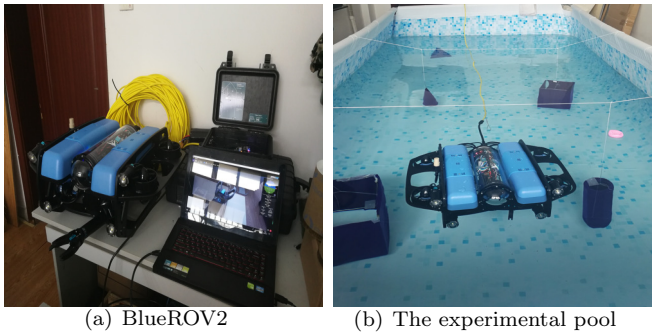


Fig. 6. Path diagram of the underwater vehicle in the simulator

the proposed method is capable of remarkable collision-avoidance for autonomous control.

Experiment The experimental hardwares mainly includes underwater vehicle and controller. Figure 7 depicts the BlueROV2 is a fully-actuated underwater vehicle developed by Blue Robotics, whose thruster configuration allows for motion in surge, sway, heave and yaw. While being equipped with a full HD resolution camera for observation purposes. An NVIDIA Jetson TX2 used to be controller, which equipped with an NVIDIA Pascal GPU is used for real-time inference and testing in reality. As shown in Fig. 8, the Jetson TX2 is a compute module from Nvidia that features a powerful and low-power CPU/GPU combo, and the compactness of the Jetson makes it ideal for the vision processing application in our experiment.

After training in simulated environment, the trained model is transferred to the real-world controller. With the raw RGB frames read using the OpenCV library, the concrete numerical values of linear velocity and angular velocity are obtained through GeoNet and CtrlNet two-stage processing. The next step, the information of linear velocity and angular velocity are transformed into the control instruction of each channel through custom communication protocol and sent to the ROV via the ground station software through UDP communication. The desired movements are achieved by controlling the speed of each thruster using corresponding motor controllers.



(a) BlueROV2 (b) The experimental pool

Fig. 7. The left side is the BlueROV2, and the right side is the experimental pool

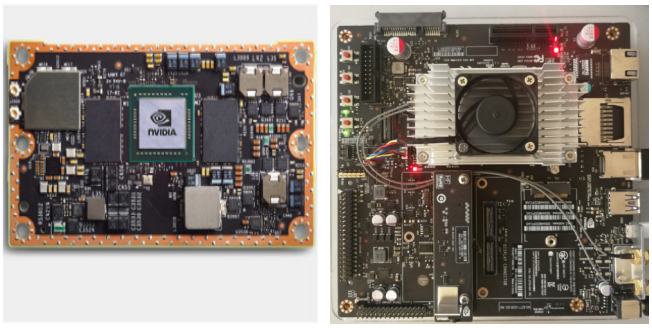


Fig. 8. Underwater vehicle ground controller-NVIDIA Jetson TX2

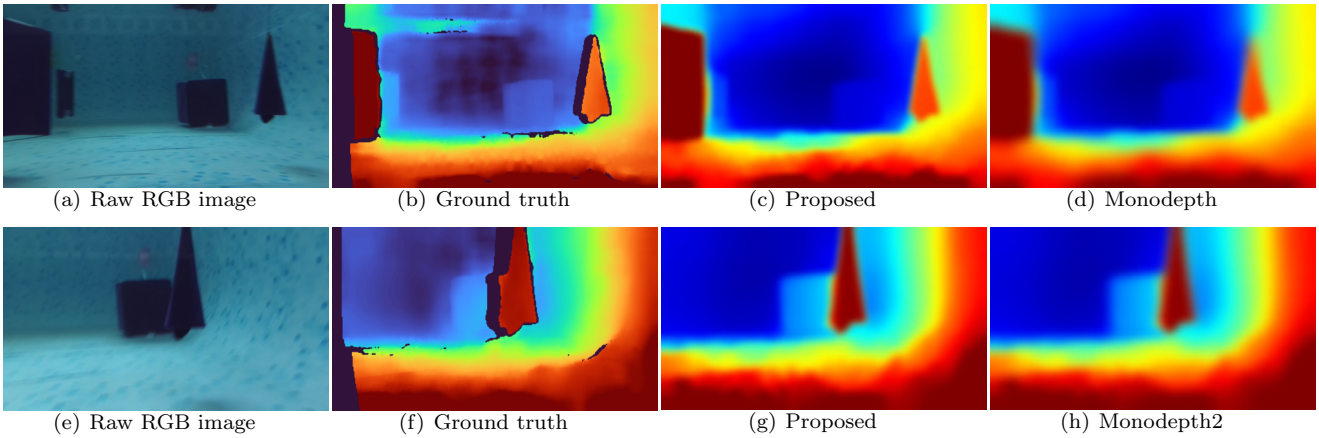
To verify the effectiveness and superiority of the monocular depth estimation, a comparison to MonoDepth (Godard et al. (2017)) is conducted. For quantitative evaluation of the monocular depth estimation, three error metrics (Eigen et al. (2014)), i.e., absolute relative difference (ARD), squared relative difference (SRD) and root mean square error(RMSE), are adopted. The results are given in Table 2, from which it is seen that the proposed method achieved best performance (shown in bold). The comparison of the visual results is shown in Figure 9, where the ground-truth depth map is inserted from the sparse measurements for visualization. From Fig.9, it is observed that the proposed method exhibits remarkable performance on depth prediction and distance feature extraction.

Table 1. Quantitative results of monocular depth estimation

Method	ARD	SRD	RMSE
Monodepth	0.162	1.578	6.104
Proposed	**0.136**	**1.029**	**5.260**

5. CONCLUSION

In this paper, a novel autonomous control framework combines depth prediction network and deep reinforcement learning is proposed by only using monocular RGB frames as input. The GeoNet is proposed for the real-time generation of depth maps and the CtrlNet is designed for autonomous control with collision-avoidance capability, which can be trained only in the simulator and then transfer directly to real-world task. Simulation and experimental results demonstrate the feasibility of transferring the



(a) Raw RGB image (b) Ground truth (c) Proposed (d) Monodepth

(e) Raw RGB image (f) Ground truth (g) Proposed (h) Monodepth2

Fig. 9. From left to right: Raw RGB images, Ground truth images, our images and monodepth images.

visual knowledge of the training network from virtual to reality, and high performance automatic control achieved by using monocular vision.

ACKNOWLEDGEMENTS

This work is supported by National Natural Science Foundation (NNSF) of China under Grant 51979021, 51479018 and 51709028, Natural Science Foundation of Liaoning province under Grant 20170520430, and fundamental research funds for the central universities under Grant 3132019317 and 3132018253.

REFERENCES

Abadi, M., Agarwal, A., Barham, P., Brevdo, E., and Zheng, X. (2016). Tensorflow: Large-scale machine learning on heterogeneous distributed systems.

Bellman, R. and Kalaba, R.E. (1965). *Dynamic programming and modern control theory*, volume 81. Citeseer.

Carlucho, I., De Paula, M., Wang, S., Petillot, Y., and Acosta, G.G. (2018). Adaptive low-level control of autonomous underwater vehicles using deep reinforcement learning. *Robotics and Autonomous Systems*, 107, 71–86.

Casser, V., Pirk, S., Mahjourian, R., and Angelova, A. (2019). Depth prediction without the sensors: Leveraging structure for unsupervised learning from monocular videos. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, 8001–8008.

Eigen, D., Puhrsch, C., and Fergus, R. (2014). Depth map prediction from a single image using a multi-scale deep network. In *Advances in neural information processing systems*, 2366–2374.

Godard, C., Mac Aodha, O., and Brostow, G.J. (2017). Unsupervised monocular depth estimation with left-right consistency. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 270–279.

He, B. and Zhou, X. (2010). Path planning and tracking for auv in large-scale environment. In *2010 2nd International Asia Conference on Informatics in Control, Automation and Robotics (CAR 2010)*, volume 1, 318–321. IEEE.

He, K., Zhang, X., Ren, S., and Sun, J. (2016). Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 770–778.

Kingma, D.P. and Ba, J. (2014). Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.

Li, Z. and Snavely, N. (2018). Megadepth: Learning single-view depth prediction from internet photos. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2041–2050.

Mahjourian, R., Wicke, M., and Angelova, A. (2018). Unsupervised learning of depth and ego-motion from monocular video using 3d geometric constraints. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 5667–5675.

Mnih, V., Kavukcuoglu, K., Silver, D., Rusu, A.A., Veness, J., Bellemare, M.G., Graves, A., Riedmiller, M., Fidjeland, A.K., Ostrovski, G., et al. (2015). Human-level control through deep reinforcement learning. *Nature*, 518(7540), 529.

Repoulias, F. and Papadopoulos, E. (2005). Trajectory planning and tracking control of underactuated auvs. In *Proceedings of the 2005 IEEE International Conference on Robotics and Automation*, 1610–1615. IEEE.

Shen, C., Shi, Y., and Buckham, B. (2016). Integrated path planning and tracking control of an auv: A unified receding horizon optimization approach. *IEEE/ASME Transactions on Mechatronics*, 22(3), 1163–1173.

Tai, L. and Liu, M. (2016). Towards cognitive exploration through deep reinforcement learning for mobile robots. *arXiv preprint arXiv:1610.01733*.

Wang, Z., Schaul, T., Hessel, M., Van Hasselt, H., Lanctot, M., and De Freitas, N. (2015). Dueling network architectures for deep reinforcement learning. *arXiv preprint arXiv:1511.06581*.

Zhang, F., Leitner, J., Upcroft, B., and Corke, P. (2016). Vision-based reaching using modular deep networks: from simulation to the real world. *arXiv preprint arXiv:1610.06781*.

Zhou, T., Brown, M., Snavely, N., and Lowe, D.G. (2017). Unsupervised learning of depth and ego-motion from video. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 1851–1858.