Proceedings of 2023 IEEE
International Conference on Mechatronics and Automation
August 6 - 9, Harbin, Heilongjiang, China

Study on 6D Pose Estimation System of Occlusion Targets for the Spherical Amphibious Robot based on Neural Network

Chaofeng Du[1],Jian Guo[1,2*]
[1]Tianjin Key Laboratory for Control Theory& Applications in Complicated systems and Intelligent Robot Laboratory
Tianjin University of Technology
BinshuiXidao Extension 391,Tianjin,300384,China
1404724876@qq.com;
*corresponding author :
jianguo@tjut.edu.cn

Shuxiang Guo[1,2,3*]
[2] Shenzhen Institute of Advanced Biomedical Robot Co., Ltd.
No.12, Ganli Sixth Road, Jihua Street, Longgang District, Shenzhen,518100,China
*corresponding author :
guoshuxiang@hotmail.com

Qiang Fu[1]
[3]Key Laboratory of Convergence Medical Engineering System and Healthcare Technology,The Ministry of Industry and Information Technology,School of Life Science
Beijing Institute of Technology
No.5, Zhongguancun South Street, Beijing,100081, China
fuqiang6369@hotmail.com

***Abstract* –The amphibious robot needs to accurately estimate the 6D pose of the target in tasks such as target tracking, docking with the recovery module, and target grasping. The current research on target 6D pose estimation is mainly applied to unoccluded targets, but when the target is occluded, the target's pose cannot be accurately identified. Compared with other algorithms, the PVNet algorithm shows better robustness when target is occluded, but the accuracy is still low. To improve the accuracy of the PVNet algorithm, this paper adds the confidence score prediction of the prediction vector at the last layer of the PVNet network, and designs a vector confidence score loss function to train the network. Before generating the hypothetical keypoints, the pixels whose confidence score is lower than the set threshold are screened out, so that the generated hypothetical 2D keypoints are closer to the true 2D keypoints. Finally, the method in this paper is compared with the Tekin, PoseCNN, Oberweger and Pvnet algorithm, and demonstrate the superiority of the proposed method.**

Index Terms –6D target pose estimation, PVNet, Amphibious robots, Semantic segmentation.

I. INTRODUCTION

One of the key technologies of amphibious robots is to be able to perceive the surrounding environment. Common 2D target detection can only provide two-dimensional plane position information and target category information of the target,but three-dimensional spatial information of the target cannot be obtained.In the amphibious robot to perform target tracking[1], recovery docking, target grasping and other tasks, the 6D pose of the target can help the amphibious robot realize the position and direction of the target for the robot's next operation and decision.Therefore, it is great significance to accurately identify the pose information of the target.

In recent years, target pose estimation methods based on deep learning have shown more computationally efficient and robust than traditional methods. The method based on deep learning has become the main research direction of target pose estimation.There are three main methods of target 6D pose estimation based on deep learning. The first is a method based on feature correspondence.In the case of knowing the complete 3D model of the target, the 2D pixel points of the input image are corresponding to it, and the 6D pose is solved from the 2D-3D correspondence. In the feature correspondence method, the most typical pose estimation networks are BB8[2] and YOLO-6D[3]. They project the vertices of the 3D bounding box onto the 2D plane to obtain the 3D-2D point-to-point relationship, and then use PnP algorithm to solve the 6D pose.The second is the 6D pose estimation method based on pixel voting. In a scene with severe occlusion, the overall pose is judged by the local feature of the target. The feature points corresponding to the 3D points or each pixel can be used to vote to obtain the 2D-3D correspondence. Among the voting methods, the most typical pose estimation networks are PVNet[4], DPVL[5] and Pix2pose[6].The principle of the PVNet network is to regress the vector field pointing to two-dimensional keypoints, and use these vectors to vote on two-dimensional keypoints based on the RANSAC algorithm. It can show strong robustness even when the target is occluded. Different from the PVNet network, the DPVL network uses the DNN(Deep Neural Network) to estimate the vector field pointing to the two-dimensional keypoints, and considers the distance between the voting pixel and the keypoints when voting. The biggest innovation of this method is propose a new loss function is used to estimate vector fields pointing to 2D keypoints. Pix2Pose, is a 6D pose estimation method for untextured targets. For untextured targets, an auto-encoder architecture is designed to estimate the 3D coordinates and the expected error per pixel. The 2D-3D correspondence is then formed using pixel-wise prediction in multiple stages to achieve the goal of directly predicting the 6D pose using PnP algorithm and RANSAC iterations.For the occlusion problem, use the Generative Adversarial Network (GAN) to accurately restore

979-8-3503-2084-8/23/$31.00 ©2023 IEEE

the occlusion part. For the symmetry problem, a new loss function is proposed for 3D coordinate regression, by guiding the predicted pose to the closest Symmetric Pose to handle pose estimation of symmetric targets. The third is the regression-based 6D pose estimation method. At present, there are many methods proposed to directly regress the position and pose of the target from the RGB image. Among the regression methods, the most typical pose estimation networks are SSD-6D[7], Pose CNN[8] and Deep-6DPose[9]. SSD-6D is to extend the 2D target detection network SSD[10] to 3D detection and 3D rotation. Pose CNN decouples 6D pose into three subtasks of semantic segmentation, translation estimation, and rotation estimation. The Deep-6DPose algorithm is to recover the 6D pose of target from a single RGB image. This framework greatly improves the efficiency by extending the segmentation network Mask R-CNN[11] and introducing a pose estimation branch to directly regress the target 6D pose without any subsequent pose refinement. Aiming at the non-differentiable and constrained problems of pose regression loss, this framework decouples the pose task into two subtasks of translation and rotation, and performs rotation regression through Lie algebra, which makes training easier, and the pose regression loss tends to close to expectations.

Due to the complex working environment of amphibious robots[12], the target is easily occluded. The 6D pose of the target cannot be accurately recognized, which will lead to the failure of the autonomous operation of the amphibious robot. At present, the PVNet 6D pose estimation algorithm shows good robustness when the target is occluded, but it can only be completed on the host computer, and it is difficult to guarantee real-time performance on the embedded side. Also, when the target is occluded, the accuracy is still low, so it is necessary to improve it.

II. The Overview of Amphibious Robot Platform

The 6D pose recognition system of occlusion targets for spherical amphibious robot consists of a spherical amphibious robot and a subsystem for 6D pose detection of occlusion targets. The hardware structure of the spherical amphibious robot is shown in fig 1. It is made up of four water jet motors, a spherical inner cabin and eight steering gears. On land, adjust the gait and attitude by controlling eight steering gears. Underwater, the underwater motion and attitude control are accomplished by controlling four water jet motors[13]. The 6D pose estimation system of occlusion targets is encapsulated in the spherical inner cabin. It consists of STM32F429 main control board, driver module, image acquisition module, power module and edge computing control board Raspberry Pi. The Raspberry Pi adopts a heterogeneous structure composed of CPU (Central Processing Unit) and GPU (Graphic Processing Unit). A 6D pose estimation algorithm is installed inside to detect the 6D pose of the target in the RGB image. The GPU is used to assist the CPU in accelerated calculation.

The specific process of the occlusion target 6D pose estimation system is: the STM32F429 main control board first transmits the RGB image collected by the image acquisition module to the edge computing control board Raspberry Pi through the UART serial port. The edge computing control board Raspberry Pi uses the internal target 6D pose estimation algorithm to estimate the 6D pose of the target in the RGB image. Then the 6D pose detection results are transmitted to the STM32F429 main control board through the UART serial port. Finally, the STM32F429 main control board controls the amphibious robot to take the next step according to the 6D pose of the target.

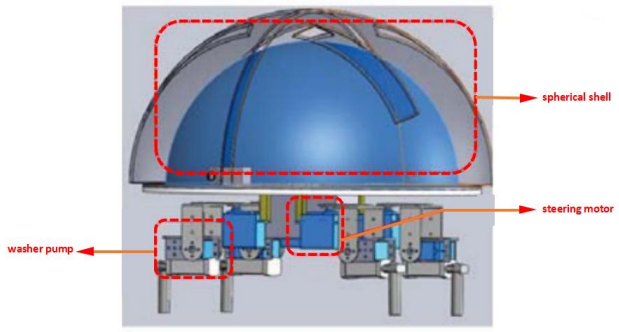


Fig. 1 Structure of spherical robot.

III. Research on 6D Pose Estimation Algorithm For Occlusion Targets

A. Network architecture

The spherical amphibious robot uses the PVNet algorithm to complete the 6D pose recognition of occlusion targets. To improve the accuracy of the 6D pose estimation of occlusion targets, this paper have improved the PVNet algorithm. Fig 2 shows the structural diagram of the PVNet network. PVNet adopts an encoder-decoder structure. The encoder layer is ResNet-18 network structure, which is used to downsample the input RGB image to extract features, and it consists of five stages. The first stage is made up of a convolutional layer and a max pooling layer. The last four stages are all composed of residual network blocks, and each stage contains two residual network blocks. And the fourth stage and the fifth stage use the atrous convolution of rate=2 and rate=4 to replace the standard convolution.The decoder layer is used to restore the feature map resolution after downsampling the RGB image for feature extraction in the encoder layer. It consists of 5 convolution layers and 3 bilinear interpolation upsampling layers.Input a picture of $H \times W \times 3$ to the PVNet network, the encoding layer downsamples the input picture to obtain a feature map with a size of $H/8 \times W/8$, and then restores the resolution of the feature map through the decoding layer until the feature The graph size is restored to $H \times W$. Finally, we apply a 1×1 convolution on the feature map to output a tensor of $\mathrm{H} \times \mathrm{W} \times (9 \times 2 + 9 \times 1 + 2)$. The prediction results of semantic segmentation, vector field pointing to two-dimensional keypoints of RGB target and vector field confidence score are obtained. Where H and W represent the height and width of

RGB images, 9×2 represents the channel occupied by the vector field in which each pixel points to 9 two-dimensional keypoints in the RGB image, 9×1 represents the channel occupied by the confidence score of the vectors pointing to 9 two-dimensional keypoints per pixel, and 2 represents the channel occupied by the semantic segmentation prediction result. Compared with the original PVNet algorithm, our proposed method adds the confidence score prediction to the vector at the last layer of the network.

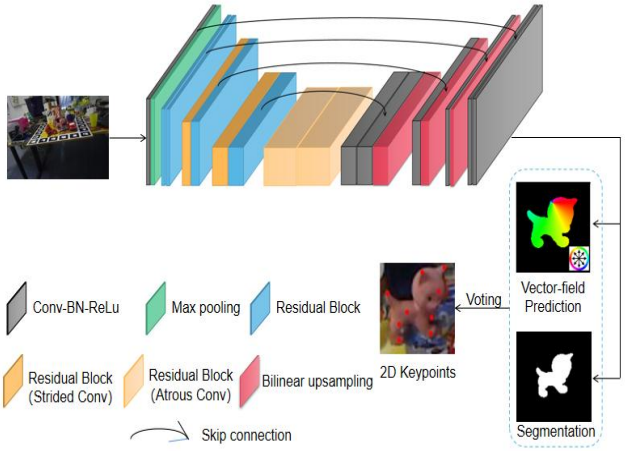


Fig. 2 The structural diagram of the PVNet network

According to the obtained semantic segmentation prediction results, the prediction vectors belonging to the pixel of the target are voted based on the random sampling consensus algorithm to obtain the coordinates of 9 wo-dimensional keypoints.Specifically: first select the pixels belonging to the target. Filter out the pixels whose vector confidence score is lower than the set threshold, and keep the pixels whose vector confidence score is greater than or equal to the set threshold. Then randomly select the prediction vectors of two target pixel points, calculate the intersection point of the two prediction vectors, and use it as the hypothetical keypoint $h_{k,i}$ for the two-dimensional keypoint x_k. By repeating this step N times, N hypothetical keypoint sets of the two-dimensional keypoint x_k can be obtained, namely: $\{h_{k,i}|i = 1,2,\cdots N\}$, and finally all the pixels belonging to the target vote for the hypothetical keypoints. If the cosine value of the angle between the direction from the pixel to the hypothetical point $h_{k,i}$ and the direction of the pixel prediction vector $\widehat{v_k}(p)$ is more than the set threshold, the weight of the hypothetical point is increased by one, and finally the coordinate of the hypothetical keypoint $h_{k,i}$ with the highest weight value is taken as the two-dimensional keypoint x_k predicted coordinates.

Using the coordinates of 9 keypoints on the 3D model of the target object and the coordinates of 9 two-dimensional keypoints projected on the RGB map of the target object, the corresponding relationship between 2D points and 3D points is obtained. Calculate the 6D pose of the target object relative to the camera through the Uncertainty-driven PnP[4] algorithm.

B. Implementation details

The 6D pose estimation algorithm PVNet for occlusion targets of the spherical amphibious robots mainly includes the following steps:

Step 1: Calculate the 3D coordinates of 9 keypoints of the 3D model of the target through the FPS (Farthest Point Sampling) algorithm, and the initial point is the center point of the 3D target.

Step 2: Use Miniconda to build the PyTorch1.1.0 environment on the upper computer. The PVNet network is trained in the PyTorch1.1.0 environment to learn the mask information of the target object projected into the two-dimensional RGB image, the vector field pointing to the two-dimensional keypoints and the confidence of the vector. After the training is completed, the model parameter file is generated.

Step 3:The PVNet algorithm and the model parameter files generated in step 2 were transplanted to the edge computing control board Raspberry Pi, and the construction of the 6D pose recognition system for occlusion targets of the spherical amphibious robot was completed.

Step 4:The image acquisition module collects the RGB image information of the working environment of the spherical amphibious robot, and transmits the RGB image information to the Raspberry Pi through the UART serial port of the STM32F429. The Raspberry Pi loads the model parameters generated in step 2 into the PVNet network to predict the collected RGB images and obtain semantic segmentation, vector fields pointing to two-dimensional keypoints, and vector confidence score prediction results. According to the obtained semantic segmentation prediction results, the prediction vectors belonging to the pixel of the target are voted based on the random sampling consensus algorithm to obtain the coordinates of 9 two-dimensional keypoints. Using the coordinates of 9 keypoints on the 3D model of the target and the coordinates of 9 two-dimensional keypoints projected on the RGB map of the target, the corresponding relationship between 2D points and 3D points is obtained. Calculate the 6D pose of the target object relative to the camera through the Uncertainty-driven PnP algorithm.

Step 5:The edge computing control board Raspberry Pi transmits the 6D pose estimation result obtained in step 4 to the main control board STM32F429 through the UART serial port for judging the next action of the amphibious robot.

C. *Design of loss function*

The PVNet network has three tasks: semantic segmentation prediction, vector field prediction, and vector confidence score prediction. Therefore, the loss function of the network consists of three parts: vector field prediction loss function, semantic segmentation loss function, and vector confidence score prediction loss function. To demonstrate the effectiveness of our proposed method, we use the same semantic segmentation loss function and vector field loss function as the original PVNet algorithm. The vector field loss function is:

$$L_{vec} = \frac{1}{n}\sum_{k=1}^{9}\sum_{p\epsilon O} l_1(\Delta \boldsymbol{v}_k(p)|_x) + l_1(\Delta \boldsymbol{v}_k(p)|_y) \quad (1)$$

$$\Delta \boldsymbol{v}_k(p) = \widehat{\boldsymbol{v}_k}(p) - \boldsymbol{v}_k(p) \quad (2)$$

Type (1): O is a collection of target pixels, l_1 is $smoothl_1$ function, $\Delta \boldsymbol{v}_k(p)|_x$ and $\Delta \boldsymbol{v}_k(p)|_y$ are the components of $\Delta \boldsymbol{v}_k(p)$ along the width and height of the image, n as the total number of pixels that belong to the target object. Type (2): $\widehat{\boldsymbol{v}_k}(p)$ is predictive vector, $\boldsymbol{v}_k(p)$ is the label vector.

The semantic segmentation loss function is:

$$L_{seg} = -\frac{1}{W\times H}\sum_{i=1}^{W\times H} p_i log q_i + (1-p_i) log(1-q_i) \quad (3)$$

Type (3): p_i is the probability of predicting pixels as target objects. q_i is the label value. W, H refers to the width and height of the RGB images.

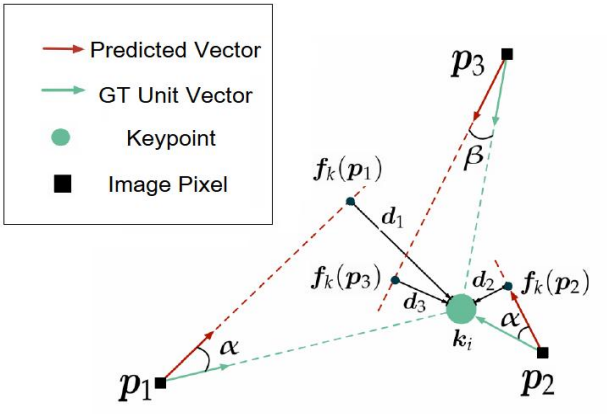


Fig. 3 Influence of voting pixel Distance on Prediction Results

As shown in the fig 3, due to the small deviation of the prediction vector of the pixel point far from the keypoint may cause a large deviation of the prediction keypoint, and the distance between a keypoint and a point on the prediction vector line is only the minimum value, there is no maximum value[5]. To solve this problem, we calculate the length of the vertical line $f_k(p)$ from the keypoint k to the prediction vector $\widehat{\boldsymbol{v}_k}(p)$ of pixel p, and normalize this length as the confidence score of the prediction vector . The shorter the length of $f_k(p)$, the higher the confidence score. The longer the length of $f_k(p)$, the lower the confidence score. In this paper, the L_2 loss function is used as the vector confidence score loss function, and the specific function is:

$$I_{cos} = cos(\widehat{\boldsymbol{v}_k}(p), \boldsymbol{v}_k(p)) \quad (4)$$

$$dis = \sqrt{(1-I_{cos})^2[(p_x-k_x)^2 + (p_y-k_y)^2]} \quad (5)$$

$$I_e = \frac{2}{\pi} arctan(\frac{1}{u_d dis}) \quad (6)$$

$$L_{score} = \frac{1}{n}\sum_{k=1}^{9}\sum_{p\epsilon O} \left\| I_e - \widehat{I_e} \right\|_2^2 \quad (7)$$

Type (4): $\widehat{\boldsymbol{v}_k}(p)$ is predictive vector, $\boldsymbol{v}_k(p)$ is the label vector. Type (5): p_x and p_y represent the horizontal and vertical coordinates of pixel point p. k_x and k_y represent the horizontal and vertical coordinates of the keypoint k. Type (6):this function is a normalized function,where u_d is the coefficient. Type (7):this function is the L2 loss function,where O is a collection of target pixels, n as the total number of pixels that belong to the target. $\widehat{I_e}$ is the vector confidence score predicted by the network, I_e is the vector confidence score label value.

V. Experiments And Results

A. *Training strategy*

This paper uses the LINEMOD[16] dataset to train the improved network model. To enhance the robustness of the model and prevent overfitting, this paper augments the training samples by rendering, cutting, resizing and rotating methods[14]. Compared with the original PVNet network, the method proposed in this paper only adds the confidence prediction of the vector field to the last layer. Therefore, to improve the training efficiency, we use the original PVNet pre-trained model to initialize the corresponding part of the network, and freeze the network parameters of all other layers except the last layer, and only update the network parameters of the last layer during the training process . In addition, the Adam optimizer is selected as the optimizer of the network, and the initial learning rate is 0.005. Every 20 epochs of training, the learning rate is halved. The batch size of each input to the network is set to 32, and our network is trained for 200 epochs. The network architecture is implemented using PyTroch 1.1.0 and trained on two Tesla V100 GPUs.

B. *Evaluation metrics*

In this paper, two general evaluation strategies are used to evaluate our method: 2D projection metric[15] and average 3D distance of model points (ADD) metric[16].

The 2D projection evaluation index is used to reflect the proximity of the 2D projection points of the 3D model of the target object using the ground-truth pose and the estimated pose respectively. Specifically, first calculate the 2D projected coordinates of the 3D model point set of the target object under the ground-truth pose and the estimated pose respectively. Then calculate the Euclidean distance between the two projected coordinates. A estimated pose is considered correct if the average distance between the 3D model projected points of the estimated pose and the ground-truth pose is less than five pixels.

ADD metric calculates the mean distance between 3D model points transformed by the estimated pose and the ground-truth pose.when the mean distance is less than 10% of the model diameter, the estimated pose is considered as correct. For symmetric objects, we use the ADD-S metric, which is different from ADD, where the average distance is computed from the nearest point distance after pose

transformation. In the remainder of this paper, for the sake of brevity, these two indicators are denoted as ADD.

C. Comparison with the state-of-the-art methods

We compared the accuracy of the proposed method with state-of-the-art 6D pose estimation methods on the LINEMOD dataset and the OCCLUSION_LINEMOD dataset.

TABLE I
2D METRIC COMPARISON ON LINEMOD DATASET

methods	BB8	Tekin	PVNet	OURS
duck	81.2	94.65	98.02	98.12
glue	89.0	96.53	98.45	98.49
ape	95.3	92.10	99.23	99.46

TABLE II
ADD METRIC COMPARISON ON LINEMOD DATASET

methods	BB8	Tekin	PVNet	OURS
cat	45.2	41.82	79.34	79.94
duck	32.8	27.23	52.58	54.41
glue	27.0	80.02	95.66	95.57
ape	27.9	21.62	43.62	44.50

TABLE III
2D METRIC COMPARISON ON OCCLUSION LINEMOD DATASET

methods	Tekin	PoseCNN	Oberweger	PVNet	OURS
cat	3.62	10.4	65.1	65.12	65.72
duck	5.07	31.8	61.4	61.44	64.12
glue	4.70	13.8	54.9	55.37	56.33
ape	7.01	34.6	69.6	69.14	70.28

TABLE IV
ADD METRIC COMPARISON ON OCCLUSION LINEMOD DATASET

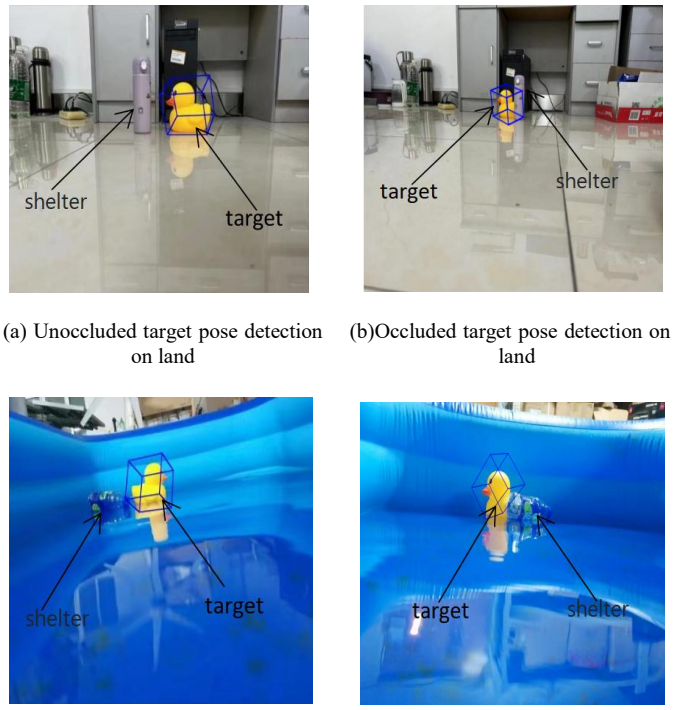
methods	Tekin	PoseCNN	Oberweger	PVNet	OURS
cat	0.67	0.93	3.31	16.68	20.17
duck	1.14	19.6	19.2	25.24	27.40
glue	10.08	38.5	39.6	49.62	50.76
ape	2.48	9.6	17.6	15.81	16.23

Firstly reproduce the BB8[2], Tekin[18], and original PVNet[4] algorithms on the Ubuntu operating system, and test them on the LINEMOD dataset. The comparison results of our proposed method with BB8, Tekin, and original PVNet on the LINEMOD dataset on 2D projection metrics and ADD metrics are shown in Table I and Table II, respectively. Experimental results shows that the proposed method is superior to BB8,Tekin and original PVNet algorithms in ADD evaluation metric and 2D projection metric.

Then we reproduce the PoseCNN[8], Tekin[18], Oberweger[19] and original PVNet[4] algorithms on the Ubuntu operating system, and we directly test on the Occlusion LINEMOD dataset[17] using models trained on the LINEMOD dataset. The comparison results of our proposed method with PoseCNN, Tekin, Oberweger and original PVNet on the Occlusion LINEMOD dataset on 2D projection metrics and ADD metrics are shown in Table III and Table IV, respectively. Experimental results shows that the proposed method is superior to PoseCNN, Tekin, Oberweger and original PVNet algorithm in ADD evaluation metric and 2D projection metric.

D. Experiments on land and water

Firstly install the 64-bit Raspberry Pi system on the edge computing control board Raspberry Pi. Connect to the display with an HDMI cable and connect to the wireless network. Then built a deep learning environment on the Raspberry Pi, and installed python3.9, pytorch1.6 and opencv packages. Finally, the PVNet algorithm and the trained network model were transplanted to the Raspberry Pi, and the 6D pose estimation hardware system for occlusion targets of the spherical amphibious robot was completed.



(a) Unoccluded target pose detection on land

(b)Occluded target pose detection on land

(c)Unoccluded target pose detection on on the water surface

(d) Occluded target pose detection the water surface

Fig. 4 Experimental results.

When the spherical amphibious robot works on water or land, the image acquisition module collects the RGB image information of the working environment, and transmits the image to the Raspberry Pi through the UART serial port of the STM32F429. The Raspberry Pi predicts the target in the RGB image through the internal PVNet algorithm, and obtains the pose prediction result. The edge computing control board Raspberry Pi transmits the 6D pose prediction result to the main control board STM32F429 through the UART serial port for judging the next action of the amphibious robot. We use the VNC host computer to connect to the Raspberry Pi, observe the 6D pose estimation results of the spherical amphibious robot to the target in real time, and obtain the experimental results. The 6D pose visualization detection results of the target are shown in fig 4, it can be observed that the amphibious robot can accurately identify the 6D pose of the occlusion target. In the picture, the Blue 3D bounding boxes represent the predicts poses.

VI. Conclusions and Future Work

This paper proposed a 6D pose estimation system of occlusion targets for the spherical amphibious robot. We used Miniconda to build the PyTorch1.1.0 environment and chose the PVNet target 6D pose estimation algorithm. To improve the accuracy of the PVNet algorithm, this paper added the confidence score prediction of the prediction vector to the last layer of the PVNet network, and designed a vector confidence score loss function to train the network. Before generating hypothetical keypoints, pixels with confidence scores lower than a set threshold are screened out, so that the generated hypothetical 2D keypoints are closer to the real 2D keypoints. Furthermore, we compared our method with the Tekin, PoseCNN, Oberweger and Pvnet algorithm, and demonstrate the superiority of the proposed method. In the follow-up work, we need to optimize the PVNet network model and voting algorithm, reduce the calculation amount of the algorithm, and improve the 6D pose detection speed of targets.

Reference

[1] S. Guo, S. Pan, et al, "A system on chip-based real-time tracking system for amphibious spherical robots," *International Journal of Advanced Robotic Systems*, vol.14, no.4, pp.1-19, Jul 2017.

[2] M. Rad and V. Lepetit, "BB8: A Scalable, Accurate, Robust to Partial Occlusion Method for Predicting the 3D Poses of Challenging Objects without Using Depth," *IEEE International Conference on Computer Vision (ICCV)*, pp.3848-3856, Oct 2017.

[3] B. Tekin, S. N. Sinha and P. Fua, "Real-Time Seamless Single Shot 6D Object Pose Prediction," *IEEE Conference on Computer Vision and Pattern Recognition(CVPR)*, pp. 292-301, Jun 2018.

[4] S. Peng, X. Zhou, et al, "PVNet: Pixel-Wise Voting Network for 6D oF Object Pose Estimation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*,vol.44, no.6, pp.3212-3223,Jun 2022.

[5] X. Yu, Z. Zhuang, P. Koniusz, et al, "6D oF Object Pose Estimation via Differentiable Proxy Voting Loss," *ArXiv*, May 2020.

[6] K. Park, T. Patten and M. Vincze, "Pix2Pose: Pixel-Wise Coordinate Regression of Objects for 6D Pose Estimation," *IEEE International Conference on Computer Vision(ICCV)*, pp.7667-7676, Feb 2020.

[7] W. Kehl, F. Manhardt, F. Tombari, S. Ilic and N. Navab, "SSD-6D: Making RGB-Based 3D Detection and 6D Pose Estimation Great Again," *IEEE International Conference on Computer Vision(ICCV)*, pp.1530-1538,Oct 2017.

[8] Y. Xiang, T. Schmidt, V. Narayanan, and D. Fox, "Pose CNN: A Convolutional Neural Network for 6D Object Pose Estimation in Cluttered Scenes," *Science and Systems (RSS),* pp.1-8, Jun 2018.

[9] Do T-T, M. Cai, T. Pham, et al, "Deep-6DPose: Recovering 6D Object Pose from a Single RGB Image," *ArXiv*, Apr 2019.

[10] W. Liu, D. Anguelov, et al, "SSD: Single Shot MultiBox Detector," *European Conference on Computer Vision(ECCV)*, pp.21-37,Sep 2016.

[11] K. He, G. Gkioxari, P. Dollár and R. Girshick, "Mask R-CNN," *IEEE International Conference on Computer Vision*, pp.2980-2988,Dec 2017.

[12] H. Xing, S. Guo, et al, "Hybrid Locomotion Evaluation for a Novel Amphibious Spherical Robot," *Applied Sciences*, vol.8, no.2, pp.1-8, Jan 2018.

[13] J. Guo, S. Guo, et al, "Design and Characteristic Evaluation of a Novel Amphibious Spherical Robot," *Microsystem Technologies*, vol.23, no.6, pp.1-14, Aug 2017.

[14] X. Hou, S. Guo, et al , "Hydrodynamic Analysis-Based Modeling and Experimental Verification of a NewWater-Jet Thruster an Amphibious Spherical Robot," *Sensors*, vol.19, no.259, Jan 2019.

[15] E. Brachmann, F. Michel, et al, "Uncertainty-driven 6d pose estimation of objects and scenes from a single rgb image," *Computer Vision and Pattern Recognition(CVPR)*, pp.3364-3372,Jun 2016.

[16] S. Hinterstoisser, V. Lepetit, et al, "Model based training, detection and pose estimation of texture-less 3D objects in heavily cluttered scenes," *Asian Conference on Computer Vision(ACCV)*, pp.548–562,Nov 2012.

[17] E. Brachmann, A. Krull, et al, "Learning 6D object pose estimation using 3D object coordinates," *European Conference on Computer Vision(ECCV)*, pp.536–551,Sep 2014.

[18] B. Tekin, S. N. Sinha and P. Fua, "Real-time seamless single shot 6d object pose prediction," *Computer Vision and Pattern Recognition (CVPR)*, pp.292-301, Jun 2018.

[19] M. Oberweger, M. Rad and V. Lepetit, "Making deep heatmaps robust to partial occlusions for 3D object pose estimation," *European Conference on Computer Vision (ECCV)*, pp.125–141, Sep 2018.