Deep Learning Based Underwater Object Detection and Pose Estimation

MyungHwan Jeon[1], Yeongjun Lee[2], Young-Sik Shin[3], Hyesu Jang[3], and Ayoung Kim[3*]

***Abstract*— In this paper, We present a deep learning based object pose estimation that specifically targets underwater applications. We improve the existing approach for making a dataset using a 3D CAD model for underwater object detection and pose estimation. We also present a fine-resolution pose estimator for underwater objects. We show that object detection and pose estimation networks trained via our dataset present preliminary potential for deep learning based approaches in underwater. Lastly, we show that our synthetic image dataset provides meaningful performance for deep learning models in underwater environments.**

I. INTRODUCTION

In recent years, deep learning methods have been utilized for recognizing objects and determining their poses and locations. Diverse deep learning method for 3D object detection and pose estimation have been proposed, with many impressive results being presented in unconditional environments and for general object recognition tasks. These methods have sufficient accuracy for robotic tasks, including the grasping of objects [1, 2]. However, implementing a deep learning based approach for underwater applications leads to two further issues.

Firstly, underwater object data acquisition itself is challenging as the dataset is scant compared to datasets for a terrestrial environment. Even after obtaining the data, manual annotation is required; this has high costs and could have inaccuracies due to human error. Secondly, the dataset can be too ambiguous to completely reflect the underwater optical environment. Underwater camera images can experience intensity degeneration and color distortion [3].

In this paper, we improve the existing approach, in which a dataset is created using a 3D computer aided design (CAD) model, to ensure that the dataset includes various optical conditions and can extract several annotations. For this research objective, we introduce an automatic annotation tool and its application to object detection and pose estimation. In addition, we propose a fine-resolution pose estimator. In experiments, we show that the object detection and pose estimation networks that are trained with our dataset present the preliminary potential for deep learning based approaches in underwater applications and validate that our synthetic image dataset presents meaningful preliminary performance for deep learning model in underwater environments.

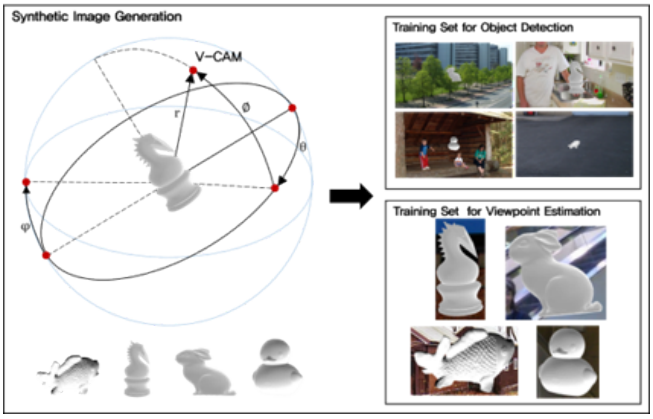

Fig. 1. Illustration of synthetic image generation. We photograph the 3D model with virtual camera (V-CAM) to create a synthetic image, and we extract the annotations for the object detection and pose estimation networks from these synthetic images. We utilize synthetic images with object mask and object class annotations for the training set of the object detection network. For the pose estimation network, We crop the synthetic images using truncation annotation. Then, we use the cropped images and pose annotations for the training set of the pose estimation network.

In summary, this paper presents three things as follows:

- We automatically generate all the necessary annotations for object detection and pose estimation.
- We propose a fine-resolution (i.e., sub-meter level) pose estimation network for underwater.
- We verify that the synthetic image set that uses a 3D CAD model is feasible for training in underwater object detection.

II. RELATED WORKS

Securing enough training data is essential for deep learning based approaches. Many researchers have focused on creating a synthetic dataset to enable automatic annotation.

A. Synthesizing Images Using a 3D CAD Model

In [4, 5], the authors utilized 3D CAD models to synthesize images. In their works, they used these images to detect objects and estimate poses. Peng et al. [4] collected 3D CAD models of a 3D warehouse by searching online for the names of 20 categories. In their paper, 25 models per category were coated with a texture, and their color was selected by the authors. During this model generation, the authors manually changed the viewpoint to render virtual images. Su et al. [5] chose 3D models from PASCAL 3D+ for the names of 12 categories. They randomly selected the

[1] M. Jeon is with the Department of the Robotics Program, KAIST, Daejeon, S. Korea myunghwan.jeon@kaist.ac.kr

[2] Y. Lee is with KRISO, Daejeon, S. Korea leeyeongjun@kriso.re.kr

[3] Y. Shin, H. Jang and A. Kim are with the Department of Civil and Environmental Engineering, KAIST, Daejeon, S. Korea [youngsik.shin, iriter, ayoungk]@kaist.ac.kr

This study is a part of the results of R&D project, Development of Basic Technologies of 3D Object Reconstruction and Robot Manipulator Motion Compensation Control, supported by KRISO.

illumination condition, viewpoint, and background to make the images. Our work is most similar to [5], but it differs from theirs by focusing on underwater implementation.

B. Automatic Annotation Tool

In order to ensure the significant performance in a deep learning based approaches, precise annotation of the target to be learned is essential in supervised manners. The most common method for addressing this issue is manual annotation. However, manual annotation causes discomfort and human-error, and it is infeasible for large datasets. To overcome this issue, researchers have used an automated annotation tool with a real image. Nevertheless, automated generation of instance annotations in images has proven challenging due to the impediment of accurate classification of instances and occlusion across objects. Using a 3D CAD model can alleviate these problems. For instance, [5, 6, 7, 8] developed automated annotation tools using 3D CAD models. Our approach automatically annotates viewpoints, bounding-boxes, and segmentation labels for use in underwater environments.

III. METHOD

To perform instance level object detection and pose estimation, we composed a cascadely connected two networks, (*i*) Mask R-CNN and (*ii*) a pose estimation network. As the first network, Mask R-CNN creates the mask and class for an object. The area where the object is located, extracted by this mask, is given as input to the pose estimation network. This mask thus allows the pose estimator to focus only on the objects for which pose is to be estimated. Secondly, the proposed pose estimation network is combined with Densely Connedted Convolutional Networks (DenseNet) [9], Dense Block and, and Fully Connected Network (FC).

A. Synthesizing the Image

The entire synthesizing phase is done automatically without human interception. As shown in Fig. 1, a 3D model is placed at the center of the spherical coordinate system. Then, the pose of the Virtual Camera (V-CAM) is changed to acquire samples for azimuth, elevation, and distance. Finally, an in-plane rotation sample is produced by rotating the plane on which the model lies, around the origin. At the same time, we obtain a transparent background image with the model centered using the V-CAM. We extract the truncation parameters and segmentation labels from this transparent background image. For the training set of the object detector, we overlay a background on to the generated transparent background image. For the training set of pose estimator, after cropping the transparent background image using the truncation parameter, we coat the cropped images with the background. This image is employed for the training set of the viewpoint estimator.

B. Pose Estimation

The proposed pose estimation network belongs to classification rather than regression. The parameters that must be classified are the azimuth, elevation, in-plane rotation, and distance (which have 360, 180, 360, and 100 bins, respectively). To approximate the four parameters with a large range, we create a new network by combining one DenseNet, four Dense Blocks and four FCs. DenseNet [9] extracts more complex key points from the learning process than the other networks do because almost all of its layers deploy the information of the previous layer through a skip connection. All parameters have shared features through the DenseNet. We assign a Dense Block to each of the four parameters; each block extracts key points for one parameter. One FC is allocated for each parameter (Fig. 2).

Networks trained in one category hardly show valuable results in other categories due to huge geometric variation of object. To address this issue, we assign a network model to each object to enhance the accuracy.

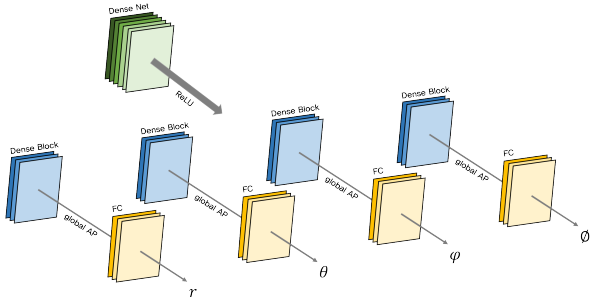

Fig. 2. Illustration of proposed pose estimator. We exploit DenseNet201. The Dense Block consists of 12 pairs of 1×1 conv and 3×3 conv. Through global Average Pooling (globalAP) and FC, the pose estimator makes four outputs.

IV. EXPERIMENTS

A. Experiment Setting

To construct our training set, we prepared four 3D CAD models. To apply our cascade networks to underwater objects, we prepare all of 3D physical models using a 3D printer. For experiments in the underwater environment, we fed in these outputs into a water tank to validate preliminary performance. (Fig. 3).

For training, we generated 1000 samples for each model to utilize the object detection, and 2000 samples were created for the pose estimation. The example results for the above process are shown in Fig. 1. We leveraged only synthetic images when object detector and pose estimator were trained. As shown in Fig. 1, we used uncropped images for the object detector and cropped images for the pose estimator.

For the test, we captured three images per object (i.e., `DUCK`, `RABBIT`, `FISH`, and `CHESS`), in differing poses, and images containing all four objects (`ALL`), as shown in Fig. 4.

B. Object Detection

In this section, we evaluate whether our training data were appropriate for underwater object detection. We obtained object detection results through the Mask R-CNN, which we trained with synthetic images. To measure the accuracy of the object detector, we used mean Average Precision (mAP)

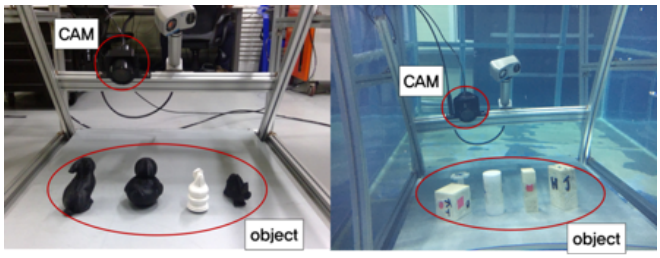

Fig. 3. Experiment setup. A camera and four objects are prepared in-air (left), and then placed in-water (right)

TABLE I

SUMMARY OF OBJECT DETECTION WITH EVALUATION METRICS.

Object	Chess	Duck	Rabbit	Fish	All
mAP	0.86	0.82	0.88	0.64	0.81
Mask Overlay	0.87	0.89	0.83	0.71	0.82

and the overrode the ratio of the ground truth mask and predicted mask (Table. I). We used the mAP as the metric for the accuracy of the object detector in the PASCAL Visual Object Class (VOC) challenge. We set the Intersection over Union (IoU) threshold to 0.5.

Except for fish, the mAP and mask overlay of the objects exceed 0.8. As shown in Fig. 5, the fish 3D model is much more detailed than the other models, resulting in the fish's scales being visible. In the training step, Mask R-CNN extracts the low-level key points using the fish's scales. However, the camera rarely captures all of the details of a fish model in underwater. For this reason, Mask R-CNN cannot utilize all the extracted key points and thus shows poor accuracy in detecting the fish model. For this reason, we obtain the undesirable result when using only the fish's

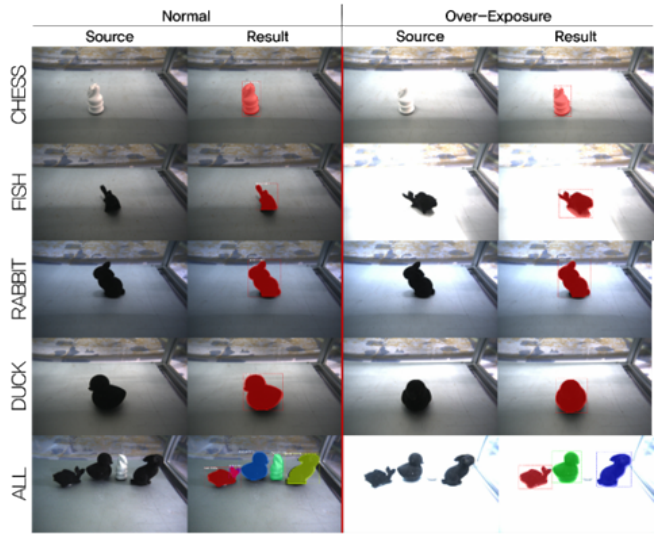

Fig. 4. Results of the object detection. The results of the four objects and the all four objects are shown. The first two columns are the source image and the detection results for the normal situation. The third and fourth columns are the detection results for the over-exposure situation. In the results column, the colored region represents detection mask.

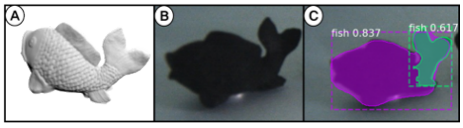

Fig. 5. Detection problem of the fish model. (A) is the 3D CAD model. (B) shows a printed model photographed in underwater. (C) is the object detection result of this model. The object class detected by our network is presented as colored regions. Detection score is presented by number.

tail as a high-level feature to detect the fish model.

The proposed method performs well for various illumination conditions. The object is reliably detected even with over-exposure situation shown in Fig. 4 without requiring preprocesses such as deblurring and dehazing.

C. Pose Estimation

Finding the ground truth of the object pose was onerous owing to the fact that both the objects and the camera were in the water tank during the experiment. Due to this issue, we made a test set using 1000 synthetic images. We used the mean absolute error between the ground truth and the predicted value to evaluate our pose estimation (Table. II). We estimated the distance using 0.1 m interval. Of the four evaluation criteria, the mean absolute error for the distance has the lowest score. However, excluding the distance, all of the mean absolute errors for object poses are relatively large. These three parameters have 360, 360, and 180 bins, respectively, but the training set we used included only 2000 images. In other words, the number of images and the diversity of dataset are scant compared to the wide range of angles used in the classification. In addition, the experiment results reveal prevalent 180° difference between the ground truth and the estimated value of the azimuth. Thus, our pose estimator could be confused when distinguishing the front and back of an object.

TABLE II

MEAN ABSOLUTE ERROR OF POSE ESTIMATION

	Azimuth	Elevation	In-plane rotation	Distance
Chess	132.43°	10.83°	4.85°	0.22 m
Duck	64.33°	12.19°	11.62°	0.17 m
Rabbit	51.11°	28.15°	13.08°	0.31 m
Fish	49.83°	26.31°	13.84°	0.28 m

V. CONCLUSIONS

In this paper, we propose the approach of making a synthetic image training set with automatic annotation using a 3D CAD model. We utilize this training set for object detection and pose estimation in underwater environment. The experiment proved that our synthetic training set present preliminary potential for underwater object detection and pose estimation. In future works, we will pursue improvements in all of the pose parameter, especially azimuth.

References

[1] X. Zhou, X. Lan, H. Zhang, Z. Tian, Y. Zhang, and N. Zheng, "Fully convolutional grasp detection network with oriented anchor box," *arXiv:1803.02209*, 2018.

[2] F.-J. Chu and P. A. Vela, "Deep grasp: Detection and localization of grasps with deep neural networks," *arXiv:1802.00520*, 2018.

[3] Z. Chen, Z. Zhang, F. Dai, Y. Bu, and H. Wang, "Monocular vision-based underwater object detection," *Sensors*, vol. 17, no. 8, p. 1784, 2017.

[4] X. Peng, B. Sun, K. Ali, and K. Saenko, "Exploring invariances in deep convolutional neural networks using synthetic images," *CoRR*, vol. 2, no. 4, 2014.

[5] H. Su, C. R. Qi, Y. Li, and L. J. Guibas, "Render for cnn: Viewpoint estimation in images using cnns trained with rendered 3d model views," in *Proceedings of the IEEE International Conference on Computer Vision*, 2015, pp. 2686–2694.

[6] H. Hattori, N. Lee, V. N. Boddeti, F. Beainy, K. M. Kitani, and T. Kanade, "Synthesizing a scene-specific pedestrian detector and pose estimator for static video surveillance," *International Journal of Computer Vision*, pp. 1–18, 2018.

[7] P. P. Busto and J. Gall, "Viewpoint refinement and estimation with adapted synthetic data," *Comp. Vis. and Img. Under.*, vol. 169, pp. 75–89, 2018.

[8] Y. Wang, X. Tan, Y. Yang, X. Liu, E. Ding, F. Zhou, and L. S. Davis, "3D pose estimation for fine-grained object categories," *arXiv:1806.04314*, 2018.

[9] G. Huang, Z. Liu, L. Van Der Maaten, and K. Q. Weinberger, "Densely connected convolutional networks." in *CVPR*, vol. 1, no. 2, 2017, p. 3.