Underwater Object Detection and Pose Estimation using Deep Learning

MyungHwan Jeon * Yeongjun Lee ** Young-Sik Shin, Hyesu Jang, Ayoung Kim ***

* Department of the Robotics Program, KAIST, Daejeon, S. Korea
(myunghwan.jeon@kaist.ac.kr)
** KRISO, Daejeon, S. Korea (leeownerjun@kriso.re.kr)
*** Department of Civil and Environmental Engineering, KAIST, Daejeon, S. Korea, (e-mail: [youngsik.shin, iriter, ayoungk]@kaist.ac.kr)

Abstract:
This paper presents an approach for making a dataset using a 3D CAD model for deep learning based underwater object detection and pose estimation. We also introduce a simple pose estimation network for underwater objects. In the experiment, we show that object detection and pose estimation networks trained via our synthetic dataset present a 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.

1. INTRODUCTION

Recently, object detection and 6D pose estimation have utilized deep learning methods. These methods have shown significantly impressive results for general object recognition tasks in unconditional environments and have sufficient accuracy for robotic tasks, including grasping objects (Zhou et al. (2018); Chu and Vela (2018)). However, the results of previous research mostly focus on terrestrial environments. Since datasets constituting underwater environment are scarce, implementing a deep learning based approach for underwater applications is challenging.

Obtaining and utilizing underwater object data lead to two further issues. Firstly, acquiring underwater object data is challenging compared to acquisition from the ground environments. Even if the data are obtained, manual annotation is costly and could have inaccuracies due to human error. Secondly, underwater camera images should have diverse variations such as intensity degeneration and color distortion (Chen et al. (2017)).

In this paper, we enhance the existing approach, in which a dataset is created by adopting a 3D computer aided design (CAD) model, to ensure that the dataset involves various optical conditions and underwater environments. To meet the research objective, we present an automatic annotation tool. Also to verify the effectiveness of our dataset, we show its application to object detection and pose estimation. In addition, we propose a simple pose estimation network. The object detection network is trained with our dataset and presents preliminary potential for deep learning based approaches in underwater applications. We verify that our dataset suits deep learning model in underwater environments.

* 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.

Fig. 1. Illustration of synthetic image generation. The 3D model is projected by a virtual camera (Virtual Camera (V-CAM)) to capture synthetic images. By extracting the annotations for the object detection and pose estimation networks from these synthetic images, we utilized these synthetic images with object mask and object class annotations for the training set of the object detection network. Then, we cropped the synthetic images using truncation annotation. These cropped images and pose annotations were piped into the training set for 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 simple pose estimation network for underwater.
- We verify that the synthetic image set using a 3D CAD model is feasible for training in underwater object detection and pose estimation.
2. RELATED WORKS

Securing enough training data is essential for deep learning based approaches. In fact, since acquiring diverse datasets is challenging, many researchers have focused on creating synthetic datasets to enable automatic annotation.

2.1 Synthesizing Images Using a 3D CAD Model

In the literature, authors have utilized synthetic images generated using 3D CAD models to detect objects and estimate poses. Peng et al. collected 3D CAD models by searching online for the names of 20 categories. In their paper, 25 models per category were coated with a texture, and the authors selected their colors. The viewpoint was changed manually to render virtual images during the model generation step. Su et al. (2015) selected 3D models from PASCAL 3D+ for 12 categories. They randomly adjusted the illumination condition, viewpoint, and background when generating the images. Our work is similar to Su et al. (2015), but we focus on underwater implementation.

2.2 Automatic Annotation Tool

Precise annotation of the target to be learned is essential in supervised manners to ensure the significant performance of deep learning based approaches. Manual annotation is the most common method of addressing this issue. However, this annotation scheme can be inaccurate, and infeasible for large datasets. Even though researchers have used automated annotation tools using a real image to overcome this issue, automated generation of instance annotations in images has been a challenging issue caused by the impediment of occlusion with other objects. A synthetic dataset can alleviate these problems. For instance, Johnson-Roberson et al. (2016) created a training set for deep learning through a highly realistic simulation engine. Through the simulation engine, changes in the weather, daytime, and nighttime were given to ensure diversity of datasets. Alternatively, Su et al. (2015); Hattori et al. (2018); Busto and Gall (2018); Wang et al. (2018) developed automated annotation tools using 3D CAD models. Our approach automatically annotates viewpoints, bounding-boxes, and segmentation labels for use in underwater environments.

3. METHOD

We composed a network comprising two cascadedly connected networks, (i) Mask R-CNN and (ii) a pose estimation network to perform instance level object detection and pose estimation. The first network, Mask R-CNN, detects mask, bounding-box, and class for an object. For input to the pose estimation network, an image is truncated by bounding-box of an object. This input truncated by bounding-box, thus, allows the pose estimator to focus only on the objects for which the pose is to be estimated. Secondly, the proposed pose estimation network was combined with DenseNet (Huang et al. (2017)), Dense Block, and FC.

![Fig. 2. Illustration of the proposed pose estimator.](image)

We exploit Densely Connected Convolutional Networks (DenseNet)121. The Dense Block consists of 12 pairs of 1×1 conv and 3×3 conv. Through global Average Pooling and Fully Connected Network (FC), the pose estimator makes four outputs.

3.1 Synthesizing the Image

The entire synthesizing stage was processed automatically without human intervention. As shown in Fig. 1, a 3D model was located in the center of the spherical coordinate system. The 3D model remained stationary, only the pose of the Virtual Camera (V-CAM) was changed to acquire samples for relative poses between the V-CAM and 3D model (i.e., azimuth, elevation, in-plane rotation, and distance). The V-CAM parameter was tuned manually. At the same time, we obtained a transparent background image with the model centered using the V-CAM for every pose sample. This transparent background image was utilized for extracting the truncation parameters and segmentation labels. For the training set of the object detection, we overlaid a background onto the generated segmentations labels. For the training set of the object segmentation, an image was generated using the truncation parameter, we coated the cropped images with the background. These images were employed for the training set of the pose estimation. Finally, we augmented all of the images using various effects to be robust in unconditional environments.

3.2 Pose Estimation

In this paper, since pose estimation was used to validate our synthetic dataset, we did not estimate full 6D poses but only the 3D orientation. Thus, we wanted to determine whether our synthetic dataset is appropriate for the pose estimation task. The proposed pose estimation network belongs to the regression task. In the rotation regression task, we used quaternion rotation representations because they do not suffer from gimbal lock, which occurs frequently in Euler angle representation, and by construction, quaternions are unit-norm, \( ||q||_2 = 1 \). The cost of quaternion regression is shown in (1).

\[
E_Q = 2 \arccos(\langle q, \hat{q} \rangle) \quad (1)
\]

\( q \) is the ground truth value, and \( \hat{q} \) is the predicted value. The operator \( \langle \cdot, \cdot \rangle \) indicates the inner product. The prerequisite of this cost is \( \langle q, \hat{q} \rangle < 0 \) to prevent quaternion ambiguity.

To approximate the four parameters in quaternion, we created a new network by combining one DenseNet, four
Dense Blocks and four FCs. DenseNet (Huang et al. (2017)) extracts more complex key points from the learning process than 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 extracted key points for one parameter. One FC was allocated for each parameter (Fig. 2).

In fact, object pose estimation from a single RGB image is known as a substantially challenging task. To solve this challenging task, we needed to build a more specific model and costs focused on orientation information like Kehl et al. (2017); Do et al. (2018); Xiang et al. (2017). Our purpose was effective evaluation of our synthetic dataset for pose estimation task, by judging if our synthetic dataset has the potential for pose estimation from simple network.

4. EXPERIMENTS

4.1 Experiment Setting

We made a training set utilizing four 3D CAD models. For the underwater experiments, we prepared 3D physical models using a 3D printer. Then, we fed these outputs into a water tank to validate the preliminary performance. (Fig. 3).

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

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

Table 1. Summary of object detection with evaluation metrics.

| Object  | Chess | Duck | Rabbit | Fish | All  |
|---------|-------|------|--------|------|------|
| AP      | 0.913 | 0.936| 0.921  | 0.865| 0.908|
| Mask Overlay | 0.916  | 0.942| 0.925  | 0.892| 0.918|

4.3 Pose Estimation

Both the objects and the camera were in a water tank during the experiment. To evaluate the pose estimation results, we made a test set using 1000 synthetic images and used the mean absolute error between the ground truth and the predicted value.

As shown in Table. 2, quaternion as the output of the pose estimation network were converted to Euler to improve
their readability. As can be seen, the roll and yaw were somewhat larger than the pitch, by nearly an average of 20°. Since the pose error and the deviation of the error between the objects were not large, we can conclude that our synthetic dataset has preliminary potential for pose estimation. On the other hand, the experimental results revealed a prevalent 180° difference between the ground truth and the estimated value of one axis. Our pose estimator could be confused when distinguishing the front and back of an object, which is a common problem in pose estimation with regression. Thus, we need to build a more specific model focused on orientation information and costs that copes with object symmetry.

Table 2. Mean Absolute Error of Pose Estimation

|       | Roll  | Pitch | Yaw  | All   |
|-------|-------|-------|------|-------|
| Chess | 27.45 | 5.85  | 37.80 | 23.70 |
| Duck  | 20.85 | 1.69  | 29.31 | 17.27 |
| Rabbit| 19.54 | 1.27  | 20.29 | 15.03 |
| Fish  | 26.94 | 2.40  | 20.64 | 16.66 |
| All   | 23.69 | 2.80  | 20.64 | 26.51 |

5. CONCLUSIONS

In this paper, we proposed an approach to making a synthetic image dataset with an automatic annotation tool using a 3D CAD model. We applied this dataset to object detection and pose estimation in underwater environment. The experiment proved that our synthetic dataset presents preliminary potential for underwater object detection and pose estimation. In future works, we will pursue improvements in the dataset for the pose estimation. Furthermore, through the pose estimation quality improvement, we seek to grasp the object in underwater environment.

REFERENCES

Busto, P.P. and Gall, J. (2018). Viewpoint refinement and estimation with adapted synthetic data. Comp. Vis. and Img. Under., 169, 75–89.

Chen, Z., Zhang, Z., Dai, F., Bu, Y., and Wang, H. (2017). Monocular vision-based underwater object detection. Sensors, 17(8), 1784.

Chu, F.J. and Vela, P.A. (2018). Deep grasp: Detection and localization of grasps with deep neural networks. arXiv:1802.00520.

Do, T.T., Cai, M., Pham, T., and Reid, I. (2018). Deep-6dpose: Recovering 6d object pose from a single rgb image. arXiv preprint arXiv:1802.10367.

Hattori, H., Lee, N., Boddeti, V.N., Beainy, F., Kitani, K.M., and Kunade, T. (2018). Synthesizing a scene-specific pedestrian detector and pose estimator for static video surveillance. International Journal of Computer Vision, 1–18.

Huang, G., Liu, Z., Van Der Maaten, L., and Weinberger, K.Q. (2017). Densely connected convolutional networks. In CVPR, volume 1, 3.

Johnson-Roberson, M., Barto, C., Mehta, R., Sridhar, S.N., Rosaen, K., and Vasudevan, R. (2016). Driving in the matrix: Can virtual worlds replace human-generated annotations for real world tasks? arXiv preprint arXiv:1610.01983.

Kehl, W., Manhardt, F., Tombari, F., Ilic, S., and Navab, N. (2017). Ssd-6d: Making rgb-based 3d detection and 6d pose estimation great again. In Proceedings of the IEEE International Conference on Computer Vision, 1521–1529.

Peng, X., Sun, B., Ali, K., and Saenko, K. (2014). Exploring invariances in deep convolutional neural networks using synthetic images. CoRR, 2(4).

Su, H., Qi, C.R., Li, Y., and Guibas, L.J. (2015). 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, 2686–2694.

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

Xiang, Y., Schmidt, T., Narayanan, V., and Fox, D. (2017). Posecnn: A convolutional neural network for 6d object pose estimation in cluttered scenes. arXiv preprint arXiv:1711.00199.

Zhou, X., Lan, X., Zhang, H., Tian, Z., Zhang, Y., and Zheng, N. (2018). Fully convolutional grasp detection network with oriented anchor box. arXiv:1803.02209.