Can underwater environment simulation contribute to vision tasks for autonomous systems?

This item was submitted to Loughborough University's Institutional Repository by the/an author.

Citation: WANG, J. ..., et al., 2019. Can underwater environment simulation contribute to vision tasks for autonomous systems? Presented at the 2nd UK Robotics and Autonomous Systems Conference, (UK-RAS 2019), Loughborough University, 24th January.

Additional Information:

- This is a conference paper

Metadata Record: https://dspace.lboro.ac.uk/2134/37230

Version: Published

Rights: This work is made available according to the conditions of the Creative Commons Attribution-NonCommercial-NoDerivatives 4.0 International (CC BY-NC-ND 4.0) licence. Full details of this licence are available at: https://creativecommons.org/licenses/by-nc-nd/4.0/

Please cite the published version.
Can underwater environment simulation contribute to vision tasks for autonomous systems?

Jiangtao Wang, Yang Zhou, Baihua Li, Qinggang Meng, Emanuele Rocco and Andrea Saiani

Abstract — To simulate the underwater environment and test algorithms for autonomous underwater vehicles, we developed an underwater simulation environment with the Unreal Engine 4 to generate underwater visual data such as seagrass and landscape. We then used such data from the Unreal environment to train and verify an underwater image segmentation model, which is an important technology to later achieve visual based navigation. The simulation environment shows the potentials for dataset generalization and testing robot vision algorithms.

I. INTRODUCTION

In the robotics field, it is a common strategy to verify applications and algorithms in a simulation environment; this is especially important for autonomous underwater vehicles (AUVs), as frequently testing the algorithms on physical prototype in real underwater environment is costly and risky due to the unpredictable environment. Furthermore, it is difficult and time consuming to collect data and generate the corresponding labels for visual-based algorithms, in particular for deep learning which usually needs high numbers of data to train a model. On the other hand, developing a realistic underwater simulation environment that is close to the real world is a laborious task.

In this paper, we developed an underwater simulation environment with Unreal Engine 4 (UE4) and, since image segmentation is an important technology to achieve visual-based navigation, used the visual dataset generated from Unreal to train a segmentation model and to verify the feasibility of using synthetic images to train underwater deep learning models. The experiment results show that the Unreal simulation environment has the capability for helping robot vision research by easily generating dataset and validating deep learning algorithms.

II. METHODOLOGY

A. Unreal Engine

The UE4 is the 4th version of Unreal Engine, and it provides useful development tool sets for building a virtual environment from enterprise applications, cinematic experiences to games on PC, mobile and so on. Hence, UE4 has the required quality to produce a realistic simulation environment for robotics research. Furthermore, the camera view in the UE4 can be controlled by keyboard and commands and there are plenty of plugins to allow developers expand UE4 functionalities. In terms of testing robot vision algorithms by communicating with the unreal environment, the UnrealCV[1] provides commands to control the UE4 for testing robot vision tasks. We can easily get four different views of the camera with UnrealCV in UE4: camera view, depth view, surface normal view and mask view.

B. Underwater environment simulation

We used UE4 to simulate the underwater environment. The environment contains seagrass, seabed, stone, rock fish and a 3D scan of a real wreck. It can simulate the light and visual conditions in the underwater environment. The views are changing varying from depth of the sea and distance towards the objects. For example, the wreck looks very hazed from distance and becomes clearer when the camera approaches to it.

C. Dataset Generation

Image segmentation dataset is collected from the simulation environment. In the UE4, we can move the camera and adjust the locations and rotations to collect a great number of image data and the segmentation labels.

D. Image segmentation

For image segmentation, we proposed a network architecture based on autoencoder network which consists of encoder and decoder networks. Our encoder network uses the architecture from VGG-16 [2] that is pre-trained on the ImageNet dataset. We add a batch normalization layer[4] and a ReLU layer[5] to each convolutional layer. The Vgg-16[4] network has 5 max-pooling layers to reduce the size of feature maps. The decoder network is the symmetric structure of our encoder network, having 5 un-pooling layers to expand the feature maps from de-convolutional layer.

Inspired by U-Net [6], we concatenate the feature maps from the last two encoders to the corresponding decoders. Moreover, we only keep concatenation operators for final two blocks of the encoder network, responsible for generating the dense feature maps, in order to reduce the training used memory.

Concatenation allows decoder network to learn from the dense feature maps by directly providing the feature maps from encoders to decoders.

Jiangtao Wang is with Loughborough University, Loughborough, The United Kingdom, LE11 3TU (e-mail: J.Wang4-18@student.lboro.ac.uk).
Yang Zhou is with Loughborough University, Loughborough, The United Kingdom, LE11 3TU (e-mail: Y.Zhou5@lboro.ac.uk).
Baihua Li is with Loughborough University, Loughborough, The United Kingdom, LE11 3TU (e-mail: B.Li@lboro.ac.uk).
Qinggang Meng is with Loughborough University, Loughborough, The United Kingdom, LE11 3TU (e-mail: Q.Meng@lboro.ac.uk).
Emanuele Rocco is with Witted Srl, Piazza Manifesta, 1 38068 Rovereto (Tn) Italy (e-mail: emanuele@witted.it).
Andrea Saiani is with Witted Srl, Piazza Manifesta, 1 38068 Rovereto (Tn) Italy (e-mail: andrea@witted.it).
III. Preliminary Results

Figure 1 shows the screenshots of four different views in the simulation environment. The lit view is camera view that shows the underwater scenery from current position. The mask view represents different objects, which will be colored with a specific color. The objects can be extracted from the mask. The normal view shows the surface of the objects. Depth view provides the relative depth of each object in current view.

![Screenshots of underwater simulation](image1.png)

- Lit (Camera)
- Mask
- Surface Normal
- Depth

Figure 1. Screenshots of underwater simulation

We get the label with a simple calculation of two images. First one is background only, and the second contains background and object. We can get the segmentation label of the objects by subtracting these two images. Figure 2 shows a training data and their label in underwater simulation segmentation dataset.

![Image Label](image2.png)

- Original Image
- Segmentation Result

Figure 2. Sample of simulation dataset

The result of segmentation algorithm that runs on simulation environment is shown in Figure 3. The seafloor, background, ship and other objects can be accurately segmented.

![Segmentation Result](image3.png)

Figure 3. Result of segmentation in Simulation data

IV. Conclusion

In this paper, we used UE4 to develop an underwater simulation environment. In this simulation environment, we can easily generate visual data for robot vision tasks and test visual algorithms. We also found that we can train the segmentation model with the dataset and labels generated from our simulation environment. In the future, a more realistic simulation that can combine with underwater dynamics is useful for further testing the visual and motion algorithms. Furthermore, since with further work the simulation environment can realistically replicate the actual underwater imagery, the model trained on the UE4 simulation could be transferred to the real world too. In future research, we would like to train a model with synthetic data and then use transfer-learning with real data to train image segmentation model for applications in real underwater environments.

V. Acknowledgement

We would like to express our thanks to Gabriele Saracco for the early development of the UE4 prototype, Ubaldo Pantaleo - ubica srl (Underwater bio-cartography) for the wreck 3D scans and Luca Fabbian for the early testing of possible ROS-UE4 integration solutions.

REFERENCES


