International Journal of Control, Automation and Systems 18(3) (2020) 523-534
http://dx.doi.org/10.1007/s12555-019-0691-3

ISSN:1598-6446 eISSN:2005-4092
http://www.springer.com/12555

Realistic Sonar Image Simulation Using Deep Learning for Underwater Object Detection

Minsung Sung, Jason Kim, Meungsuk Lee, Byeongjin Kim, Taesik Kim, Juhwan Kim, and Son-Cheol Yu*

Abstract: This paper proposes a method that synthesizes realistic sonar images using a Generative Adversarial Network (GAN). A ray-tracing-based sonar simulator first calculates semantic information of a viewed scene, and the GAN-based style transfer algorithm then generates realistic sonar images from the simulated images. We evaluated the method by measuring the similarity between the generated realistic images and real sonar images for several objects. We applied the proposed method to deep learning-based object detection, which is necessary to automate underwater tasks such as shipwreck investigation, mine removal, and landmark-based navigation. The detection results showed that the proposed method could generate images realistic enough to be used as training images of target objects. The proposed method can synthesize realistic training images of various angles and circumstances without sea trials, making the object detection straightforward and robust. The proposed method of generating realistic sonar images can be applied to other sonar-image-based algorithms as well as to object detection.

Keywords: Forward scan sonar, GAN, generative adversarial network, sonar imaging, sonar simulator.

1. INTRODUCTION

A challenge of developing and utilizing various sonar-based algorithms [1–5] is that acquiring sonar images is difficult. For example, sonar-based object detection methods in [6–8] requires thousands of images capturing a target object to train the neural networks (NN). Unfortunately, there exist few open-source sonar image datasets, because imaging sonars are not popular with the public. Manual experiments are required to acquire sonar images; however, underwater experiments are difficult and time-consuming.

Instead, some sonar simulators have been developed to synthesize sonar images [9–12], and they have been utilized to develop new methods recently. Joe *et al*. [13] used simulators of two different sonar sensors for developing their sonar-fusion-based mapping algorithms. Kim *et al.* [14] simulated sonar images of various shapes of objects to verify their three-dimensional (3D) reconstruction algorithms.

However, implementing a realistic sonar simulator is difficult. Devising a perfect mathematical model of the acoustic beam is hard because there are a variety of phenomena that affect the propagation of the acoustic beam, such as multipath reflection, backscattering, and reverberation [15,16]. Even if it modeled, it is still computationally heavy because those phenomena should consider many parameters such as beamforming, conditions of medium, surrounding terrains [17].

We herein proposed a method to simulate a realistic sonar image through two steps. First, by emulating the imaging mechanism of the sonar sensor, the proposed method simulated simple sonar images that contain only essential semantic information such as highlight and shadow of the viewed scene. Then, generative adversarial network (GAN) generates more realistic sonar images from the simulated image by adding degradation effects. This deep-learning-based approach can find a mapping between semantic information and real sonar image, which is hard to model.

Another challenge utilizing sonar-based algorithms is that sonar images have low quality. As a result, it is hard to extract features such as corners and edges from the sonar images, and algorithms using sonar images have limited accuracy.

The proposed method can also translate real sonar images into simulated-like images by training the GAN swapping the input and label. The simulated images contain important semantic information such as highlight, shadow, and background. Therefore, the translated images

Manuscript received August 20, 2019; revised November 12, 2019; accepted November 14, 2019. Recommended by Associate Editor Jinwhan Kim under the direction of Editor Keum-Shik Hong. This work was supported by the National Research Foundation of Korea(NRF) grant funded by the Korea government(MSIT) (No. 2017R1A5A1014883).

Minsung Sung, Jason Kim, Meungsuk Lee, Byeongjin Kim, Taesik Kim, Juhwan Kim, and Son-Cheol Yu are with the Department of IT Engineering, Pohang University of Science and Technology (POSTECH), 77, Cheongam-ro, Nam-gu, Pohang-si 37673, Korea (e-mails: {ms.sung, js21kim, meungsuklee, kbj0607, weed3450, robot_juhwan, sncyu}@postech.ac.kr).
* Corresponding author.

can be helpful to extract more reliable information from the sonar images and can be applied to improve the accuracy of sonar-image-based algorithms.

We applied the proposed method to underwater object detection. Underwater target detection is an essential technique to automate underwater operation [18–21], but hard to implement because of the difficulty of acquiring data of the target as well. On the other hand, a NN trained only with the proposed images without real sonar images could detect the target during a sea trial. Therefore, the proposed method can help to develop robust object detection more efficiently because the proposed method generated realistic sonar images of the target object under desired conditions in a short time. The proposed method can be applied to other sonar-image-based algorithms as well as to object detection.

This paper is organized as follows: Section 2 explains previous works to implement realistic simulators for the sonar sensors. In Section 3, we describe the proposed method to generate realistic sonar images using the ray tracing and GAN. Section 4 presents the experimental results of the proposed method. This paper ends with a conclusion in Section 5.

2. RELATED WORK

The sonar simulator has been developed to calculate how an object will appear in an image more easily. One of the most basic methods to implement the sonar simulator is using ray tracing [9, 10]. However, these simulators approximated acoustic beams, so there are some differences between simulated images and real sonar images.

More realistic sonar images can be simulated by modeling other acoustic phenomena. Kim *et al.* [14] added speckle noise, which is a typical noise of a sonar sensor. Riordan *et al.* [11] and Cerqueira *et al.* [12] took account of phenomena related to interferences and scattering. However, simulating a sonar image requires to calculate which beams affect the pixel, how the beams propagate, and how other beams interfere in, for every pixel. So, as a new parameter increases, the computation becomes sharply heavy. To tackle this problem, they used General-Purpose computation on Graphics Processing Units (GPGPU) or rasterization in the simulations. Still, these approaches may be hard to apply to an underwater environment where computational power is limited. Moreover, some phenomena, such as crosstalk and multipath, are challenged to be modeled according to the surrounding environments [15].

Some researchers have proposed to produce realistic sonar images by applying neural style transfer (NST) to prior information. NST is a method to find image-to-image mapping using a NN. From a large amount of dataset, NNs having deep architecture can find a high-dimensional and non-linear mapping, which is hard to model. Lee *et al.* [22] generated realistic sonar images of divers by applying a convolutional neural network (CNN) to a depth map of the simulated scene. Chen *et al.* [23] also applied CNN for style transfer to manually annotated semantic maps to produce sonar images of target objects placed on the seabed of the desired type.

We proposed a method to generate realistic sonar images using the simple ray-tracing-based simulator and GAN. We modeled the sonar simulator to calculate only the semantic information of the viewed scene for the fast calculation. Then, the GAN generated realistic sonar images by adding degradation effects, such as noise and blurred edge, to the simulated images.

The proposed method has three advantages. First, the ray-tracing-based simulator can provide accurate prior information considering imaging principles of the sonar to the GAN. Moreover, the GAN of the proposed method was trained for the degradation effect so that it can process images for a more general scene. Finally, the GAN trained in the reverse direction can also generate semantic information of real sonar images by removing the degradation effect and provide more reliable information of the scene. The next section describes the proposed method in more detail.

3. REALISTIC SIMULATION USING RAY TRACING AND GAN

3.1. Ray-tracing-based simulation

We first analyzed an imaging mechanism of a sonar sensor to implement the sonar simulator. A sonar sensor consists of transmitters and receivers. The transmitter projects a fan-shaped acoustic beam with a vertical beam angle ϕ. The receiver then measures the time-of-flight and intensity of the returned beams like Fig. 1(a). The early returned beams form the lower part of the image, and the pixel value becomes bright if the intensity is high. By repeating transmit and receive for different azimuth angles, a sonar image identifies the scene inside the field of view (FOV) of the sonar sensor like Fig. 1(b).

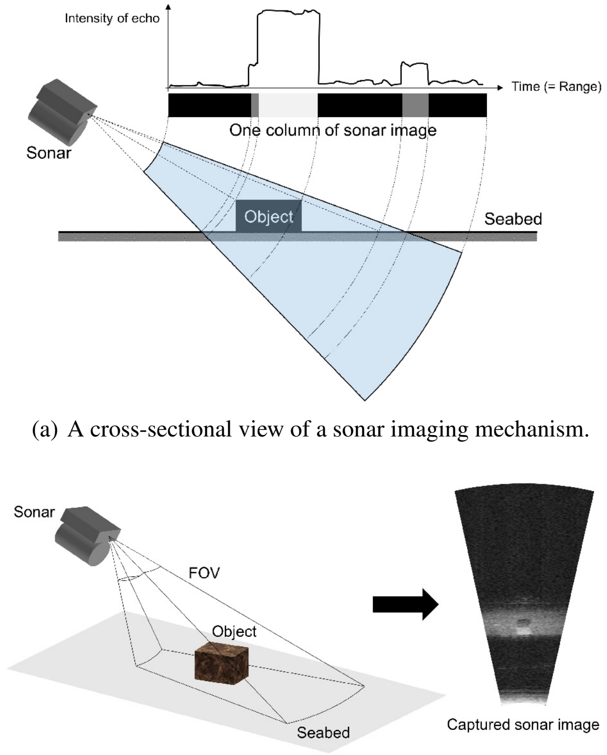
Sonar images can be simulated by emulating the imaging mechanism of the sonar sensor using the ray-tracing algorithm [24]. We first modeled a fan-shaped beam with a vertical angle ϕ as a collection of discrete K rays like Fig. 2. Then, a point p that lies on a projected ray is represented as:

$$\vec{p} = t \cdot \overrightarrow{v_{\theta,k}}, \tag{1}$$

where t is constant, $\overrightarrow{v_{\theta,k}}$ is a unit direction vector of the *sample ray*$_{\theta,k}$.

Then, we calculated the reflection point $\overrightarrow{p_{\theta,k}}$ between the ray and the underwater objects as follows:

$$\overrightarrow{p_{\theta,k}} = \frac{\vec{N} \cdot \vec{p}_1}{\vec{N} \cdot \overrightarrow{v_{\theta,k}}} \overrightarrow{v_{\theta,k}}, \tag{2}$$



(a) A cross-sectional view of a sonar imaging mechanism.

(b) Capturing a sonar image for a viewed scene.

Fig. 1. Imaging mechanism of the sonar sensor.

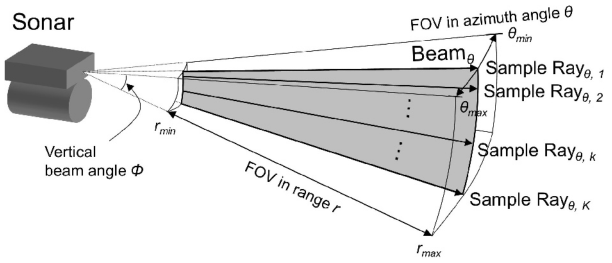


Fig. 2. Acoustic beam modeling of the sonar simulator.

where $\vec{N}$ is the normal vector of the surface of the object, $\vec{p_1}$ is one of the position vectors of vertex of the surface.

Then $\overrightarrow{p_{\theta,k}}$ is tested if it is within the measuring range of the sonar sensor using two conditions,

$$r_{min} < |\overrightarrow{p_{\theta,k}}| < r_{max}, \tag{3}$$

$$\theta_{min} < \theta < \theta_{max} \tag{4}$$

where r_{min}, r_{max}, θ_{min}, and θ_{max} are minimum and maximum measuring range of the sonar sensor in the distance and azimuth angle, respectively.

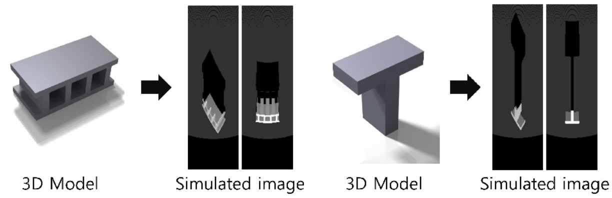


Fig. 3. Sonar image simulation based on the ray tracing.

The reflected acoustic wave from the surface spreads out in all directions and reaches the receiver of the sonar sensor. Many sonar phenomena affect the intensity of the returned beam. However, we modeled minimal acoustic phenomena, transmission loss according to the travel distance and incidence angle according to Lambert's cosine law [25], to calculate the essential information in a short time. We also assumed that the propagation and reflection of the acoustic rays are ideal. As a result, the intensity from the reflection point $I_{p_{\theta,k}}$ is calculated as:

$$I_{p_{\theta,k}} = w\frac{I_0}{|\overrightarrow{p_{\theta,k}}|^2}\cos^2\alpha, \tag{5}$$

where w is the unit conversion constant, I_0 is the initial intensity of the acoustic wave, and α is the angle between the ray and the surface.

The intensity of the reflected ray determines the pixel intensity of the image. The pixel value is calculated as follows, considering the rays affecting the pixel:

$$I(r,\theta) = \sum_{k=1}^{K} I_{r,\theta,k}, \tag{6}$$

for $r_{min} < r < r_{max}$, $\theta_{min} < \theta < \theta_{max}$, where $I(r,\theta)$ is a pixel value of (r,θ), and $I_{r,\theta,k}$ is intensity of the echo by the *sample ray*$_{\theta,k}$ from the distance r.

The reflection of the sample ray does not occur until it reaches the reflection point, and the sample ray does not travel beyond the reflection point because the object blocks it. So, the intensity of the returned ray $I_{r,\theta,k}$ is represented as:

$$I_{r,\theta,k} = \begin{cases} I_{p_{\theta,k}}, & \text{if } r = |\overrightarrow{p_{\theta,k}}|, \\ 0, & \text{otherwise.} \end{cases} \tag{7}$$

Finally, the sonar image of a given 3D model is simulated like Fig. 3 by mapping all the calculated pixel values on the 2D coordinate and applying the normalization as a final step.

3.2. Realistic simulation using GAN

The simulated sonar images are different from the real sonar images, as shown in Fig. 4. Equations (5) and (7) assume acoustic beams are propagated and reflected ideally. Moreover, other sonar phenomena such as refraction, reverberation, and multipath reflection are not considered. As a result, the simulated images are noise-free and simplified versions of the real sonar images. Calculating the

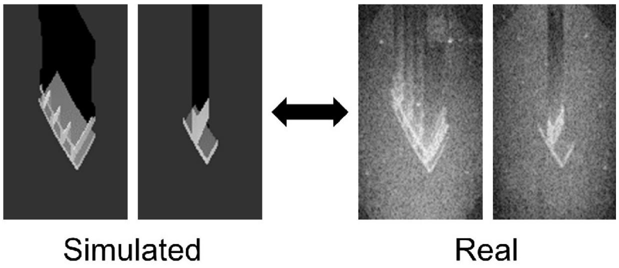


Fig. 4. Comparison of the simulated images and real sonar images.

intensity of the returned ray by modeling those sonar phenomena can simulate more realistic sonar images. However, the modeling is difficult, and the calculation is complicated and takes a long time, even if it modeled.

Instead, we proposed a method of using deep learning. Although the images simulated by the ray tracing is simple and different from the real sonar image, it contains accurate semantic information of the scene. Therefore, we can use these images as a base image and generate a realistic sonar image by applying the NST. Recently, GAN has demonstrated outstanding performance in synthesizing realistic images with desired features. Thus, the GAN for the NST was introduced for the proposed method.

We proposed a method to solve both problems of generating realistic images from the simulated images (sim-to-real translation) and generating denoised and ideal images from real sonar images (real-to-sim translation). Among the GANs developed for the NST, we adopted the “Pix2pix network” proposed by Isola *et al.* [26]. The pix2pix network has a proper structure for both translations with single network architecture. This network also has an advantage in processing sonar images that have lots of degradation effects such as blurred edges and severe noise.

The generator of the GAN is a U-Net [27] with 15 layers. U-Net has the encoder-decoder structure. The encoder extracts features from the given images, and the decoder reconstructs new images of the target domain using the extracted features. Because the encoder builds contextual information by extracting and pooling the features through multiple layers, the generator can transfer the style of an image to others preserving important information such as highlight and shadow region of the input image.

Besides, the generator has the skip-connections between nth layers and $(16-n)$th layers. By copying the feature maps of earlier layers through these skip-connections, the generator can localize the contextual information of the input images more accurately in the output images. Accurate localization within the sonar images is essential because sonar images are used to identify the range and azimuth angle of terrains in underwater exploration.

Finally, the U-Net has been initially developed for the segmentation of the images. Real-to-sim translation can be regarded as a segmentation removing degradation effects and finding regional information of highlights and shadows from the real sonar images. Therefore, the generator is appropriate for both sim-to-real and real-to-sim translations.

The discriminator is the CNN consisted of four convolutional layers. In the GAN, the discriminator distinguishes whether the input image is a real image or an image generated by the generator. As a result, CNN is used for the discriminator, which has shown remarkable accuracy in various classification problems. The discriminator observes the input image in units of patches through convolutional operations. Therefore, it makes the generator represent the details of the image better when synthesizing the images.

We designed the loss function of the network to make the GAN process sonar images of general scenes. The loss function of GAN for NST, which translates only the style of the given image preserving contextual information, is represented as:

$$\begin{aligned} Loss_{base}(G,D) =& \mathbb{E}_{x,y}[\log D(x,y)] \\ &+ \mathbb{E}_{x,z}[1-\log D(x,G(x,z))], \end{aligned} \tag{8}$$

where D is the discriminator, G is the generator, x is the given input image, y is the real image, and z is a random input vector. The generator predicts the output from the base image, and the discriminator observes the base image simultaneously to check whether the generated images contain the contextual information well.

We added a loss term for the generator to make the network learn the degradation effect of the sonar images rather than the contents, expressed as:

$$Loss_{DE}(G) = \mathbb{E}_{x,y,z}[||\mathcal{N}(y-x) - \mathcal{N}(G(x,z)-x)||_1], \tag{9}$$

where $\mathcal{N}(x)$ is a normalize function that maps x to $[-1,1]$. With this loss term, the generator is trained to find the degradation effects that can be added to the simulated ideal image to make the simulated image look like a real image. Thus, the network can make the sim-to-real and real-to-sim translation more independently of the scene of the input images.

In conclusion, the final loss function of the GAN is

$$Loss_{GAN}(G,D) = Loss_{base}(G,D) + \lambda Loss_{DE}(G), \tag{10}$$

where λ is a weight for controlling the ratio of two loss functions. Then, the generator is trained to minimize the loss value, while the discriminator tries to maximize it.

3.3. Application to underwater object detection

As one application of the proposed method, we present an underwater target object detection like Fig. 5. We synthesized realistic sonar images of the target object using

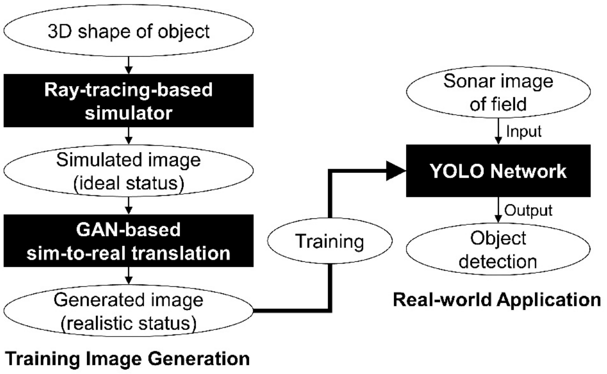

3D shape of object
Ray-tracing-based simulator
Simulated image (ideal status)
GAN-based sim-to-real translation
Generated image (realistic status)
Training Image Generation
Training
Sonar image of field
Input
YOLO Network
Output
Object detection
Real-world Application

Fig. 5. Object detection pipeline using the proposed method.

the proposed sim-to-real translation and detected the target object in the images of the field by training a CNN with the generated images. First, we model the 3D shape of the target object. Then, the ray-tracing-based simulator simulates images of the semantic information of the target object. Images are simulated while changing the yaw angle of the 3D model in n degree increments to detect the target object robust to the viewpoint. We can also change the tilt angle of the virtual sonar sensor according to a predefined set. Then, the GAN generates realistic images of target objects from simulated images of various viewpoints. Finally, CNN for object detection is trained with the generated realistic images.

Among various methods for object detection, we introduced the deep neural network (DNN). Several methods for object detection have been developed for robot vision [28, 29] and underwater scenes [6–8, 20], and recently, the DNNs have shown high detection accuracy in the sonar image. We adopted "You Only Look Once (YOLO) [30]" network. The YOLO network is a single CNN that simultaneously predicts the bounding box and class probability of object candidates. Therefore, it shows less computation and fast processing speed, and it is suitable for underwater operation in which computational power and operation time are limited. Moreover, despite the high processing speed, the YOLO network records high detection accuracy.

Using the proposed method, we can implement object detection efficiently. The proposed method uses only generated sonar images without real sonar images to train the CNN. Because synthesizing sonar images through the sonar simulator and sim-to-real translation requires less time and effort compared to capturing real sonar images, we can develop object detection more conveniently. Moreover, we expect that the proposed method can improve the detection accuracy and robustness. The sonar simulator can synthesize sonar images in a variety of circumstances that can be difficult to reproduce in real-world experiments, such as biofouling is in progress or part of the object is buried in the seabed. Training the CNN with these images can help to improve the robustness of object detection.

4. EXPERIMENT & RESULT

4.1. Training of the GAN

We first constructed a dataset to train the GAN. The dataset consists of real sonar images capturing the underwater scene and their corresponding simulated images. In the case of sonar images of the sea, there are many natural terrains, such as stones and seaweeds. Moreover, the experimental conditions, such as changes in water temperature and the appearance of moving objects, is difficult to control. As a result, modeling and simulating the environment of the sea is hard to feasible. Therefore, we conducted indoor water tank experiments to capture real sonar images and obtained corresponding simulated images by emulating the same condition.

Indoor water tank experiments were designed to capture large numbers of and various sonar images, like Fig. 6. To make the GAN generate a more realistic image, a large number of training images are helpful. Moreover, to make the GAN process more general sonar images such as those from the unknown scene, various forms of sonar images are required.

In sonar images, the shape of an object can vary considerably according to viewpoints due to the imaging mechanism of the sonar sensor. Addressing this fact, we designed a turntable to capture diverse shapes of sonar images efficiently. The turntable consists of the stepping motor at the bottom and the board on the motor. It can rotate the objects on the board to the desired angle. The board is made up of wood to distinguish it from the material of the objects. In the experiments, we can obtain the various sonar images rotating the objects by five degrees increment with this turntable.

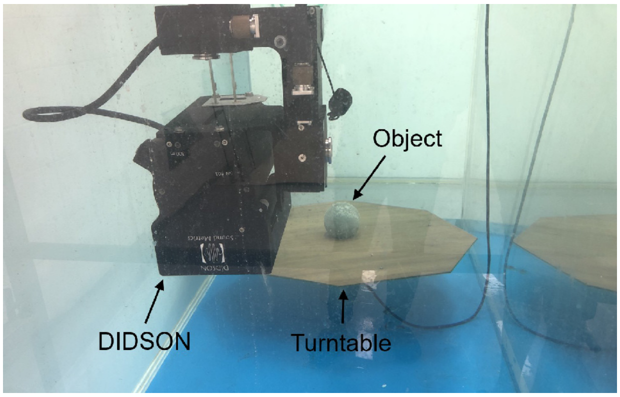


Fig. 6. Experimental setup for capturing real sonar images of various objects and viewpoints.

Moreover, by changing the position where to place objects on the board, we could change the position of the objects within the sonar image. The geometrical translation is an often-used technique for data augmentation. However, because the coordinate of a sonar image is different from an optical image, multiplying the translation matrix by the pixel value of the sonar image may not be an accurate geometrical translation. So, we acquired sonar images that the object appeared in various positions by moving the object on the turntable and then rotating the turntable.

Next, ten objects of different shapes and materials were used in the experiments. Fig. 7 shows examples of the objects. Besides, we used a concrete sphere, plastic sphere-shaped container, rubber tire, clay bricks stacked in the shape of an ‘E,’ ‘L,’ and ‘N.’

Finally, we also changed the tilt angle of the sonar to make the viewpoint more diverse. For the sonar sensor, we used the “Dual-frequency Identification Sonar (DIDSON)” [31]. Table 1 explains the experimental settings, and Table 2 describes the specifications of the DIDSON.

We then synthesized simulated images that correspond to these real sonar images using a ray-tracing-based simulator. First, we set parameters of the simulator according to settings of indoor experiments and specifications of DIDSON, as shown in Table 3. Then, like Fig. 7, we modeled 3D shapes of the object used in the experiments with the same dimensions. Finally, we placed the virtual sonar sensor in the same conditions as the experimental conditions using the translation matrix, and then simulate the images multiplying the rotation matrix to the 3D model.

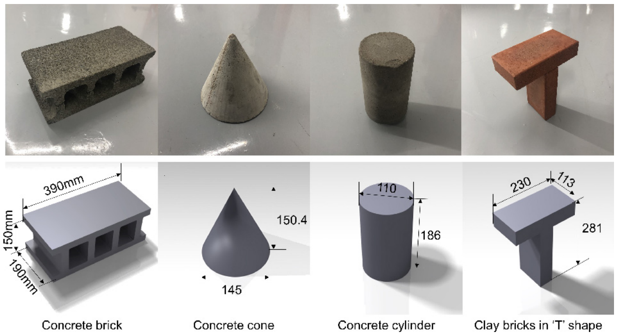


Fig. 7. Real objects and modeled 3D shapes used to construct the dataset.

Table 1. Settings for indoor water tank experiments to acquire real sonar images.

Parameter	Value
Tank size (width×length×height)	1.35 m×3 m×1.7 m
Sonar position *x*, *y*, *z* (from the center of the turntable)	1.6 m, 0 m, 0.9 m
Sonar tilt	15 °, 20 °, 25 °
Object translation	0 m, 0.15 m

We pre-processed the image pairs to train the GAN more effectively. Because the acoustic beam of the sonar sensor has a limited vertical width, there exist areas where echo is not formed in the sonar images. Furthermore, in the tank, the background of the image looks monotonous because there is no terrain on the floor. Because those shadow areas and monotonous floors do not contain much semantic information of the scene, they make computation complex and disturbs the GAN from training. Therefore, we manually cropped the area around the object that contains the most information for each image.

To construct the dataset with more diverse images, we also applied data augmentation. Because some of the objects used in the experiment have a symmetric shape, they may appear similar even though they rotated. Thus, we picked out similar images, observing every image pairs manually. Then, we transformed the object in the images into random scales and ratios.

Finally, we resized the image pairs into 256 by 256 and constructed the dataset of 2,404 image pairs. Among these, we used 2,224 pairs as training images and 180 images as test images. We separated the dataset so that the object type or viewpoint did not overlap between the training images and the test images. Fig. 8 shows the samples of training image pairs.

Using these real sonar images and their corresponding simulated images pairs, we trained the sim-to-real and real-to-sim network models. With the proposed GAN, we could train both models by swapping input and label images of the GAN. For the sim-to-real translation model,

Table 2. Specifications of the DIDSON.

Parameter	Value
Operating frequency	1.8 MHz
Vertical beam angle	14 °
Azimuth field of view	29 °
Range field of view	12 m
Maximum resolution	0.3 °
Image size	512×96
Frame rate	4 - 21 FPS
Depth rating	300 m

Table 3. Parameters of the sonar simulator to emulate DIDSON.

Parameter	Value
Min range r_{min}	0.42 m
Max range r_{max}	2.50 m
Min azimuth angle θ_{min}	-14.5 °
Max azimuth angle θ_{max}	14.5 °
# of vertical sampling rays K	1,000
Image size to simulate	512×96

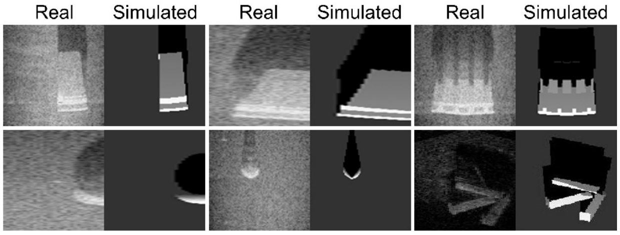


Fig. 8. Training images for GAN composed of the real sonar images and their corresponding simulated images.

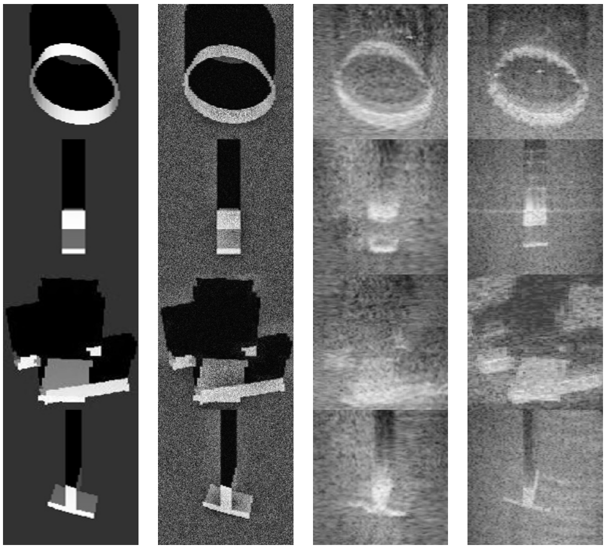
we used the simulated images as inputs and the corresponding real sonar images as labels of the GAN. For the real-to-sim translation model, we used the real sonar images as inputs and the corresponding simulated images as labels. Classically, when training a network for segmentation, constructing a dataset is difficult and time-consuming because every pixel of each image must be manually annotated to which class that pixel belongs. In the proposed method, however, we can easily create a semantic map of the scene with the ray-tracing-based simulator.

4.2. Experimental result

With the training dataset, the sim-to-real model was trained for 200 epochs, and the real-to-sim model was trained for 80 epochs. We utilized the GPU Titan V for the training, and the training took about three hours and an hour, respectively. During the training, the generator and discriminator operated adversarial to each other. Initially, the accuracy of the discriminator increased as the generator synthesized significantly different images from the target domain. The generator then produced more sophisticated images to fool the discriminator that became more accurate. As a result, the GAN generated more and more realistic and target-like images as the training progressed.

After the training was completed, we tested the trained models with a test set consisting of images of unknown objects or unknown viewpoints. We first tested the sim-to-real model. To evaluate the performance of the proposed method to generate a realistic sonar image, we implemented another method to simulate a realistic sonar image by adding modeled speckle noise to the ray-tracing-based simulator and compared the results. Fig. 9 shows the result images generated by the proposed method. When the simple images like Fig. 9(a) are input, the GAN generates images, as shown in Fig. 9(c), similar to corresponding ground truth (GT). Compared with the results by the speckle-noise-adding method of Fig. 9(b), the proposed method represents the degradation effect more detail and is more similar to GT in the distribution of the overall pixel intensities.

For the quantitative evaluation of the proposed method, we measured the similarity of the generated images with the GT. 2D discrete cross-correlation and structural similarity (SSIM) [32] were used to calculate the similarity between two images. 2D discrete cross-correlation calculates the similarity based on each pixel intensity, and SSIM shows context-based similarity. We compared the similarities of the images generated by the proposed method and of the images generated by the speckle-noise-adding method with the GT. Table 4 shows the results. The proposed method recorded the highest similarity in both metrics. It shows that the proposed method can generate realistic sonar images of the target objects.



(a) Input (b) Speckle (c) Proposed (d) GT

Fig. 9. Results of realistic sonar image generation using the proposed method.

Table 4. Similarity values of the simulated images.

	Ray tracing	Ray tracing +Speckle noise	Ray tracing +Sim-to-real translation
Cross-correlation	0.5367	0.3699	**0.6639**
SSIM	0.2082	0.0485	**0.2704**

In the case of images adding speckle noise, the similarity was even lower than the similarity of images simulated by the simple ray tracing, even though it seemed realistic to the human eye. It might be because the sonar images in the real world have more complex forms of degradation effects. Although the speckle noise is one of the noises caused in the sonar sensor, the speckle noise is not sufficient to simulate highly realistic images and utilize them as real sonar images, unless additional sonar phenomena are modeled mathematically and the parameters of those models are predicted accurately.

We then measured the processing speed of the proposed method. The ray-tracing-based simulation took 0.217 seconds, and sim-to-real translation took additional 0.066 seconds when using the GPU Titan V to generate one image. As a result, the proposed method can generate 3.53 images per second. Because the proposed method can generate realistic sonar image in a very short time when compared to performing experiments manually, it can be utilized to simulate images to develop various sonar-image-based algorithms.

We then tested the real-to-sim translation model. We also used the test dataset, including sonar images of the unknown object or the unknown viewpoints. Fig. 10 shows the result of the generated semantic map using the real-to-sim translation. Given a real sonar image, GAN translated the input image into a denoised, segmented form which has only highlights, backgrounds, and shadows similar to the GT. As a result, by applying the thresholding to real-to-sim results as post-processing, we could generate a semantic map of the captured scene.

For the quantitative evaluation of this model, we measured Peak Signal to Noise Ratio (PSNR) and segmentation Intersection over Union (IOU). Through the PSNR, we can verify whether the degradation effects of sonar images have been effectively removed. The PSNR increased by about 16 dB to 39.11 dB after the proposed method from 22.92 dB before the real-to-sim translation. Then we measured the segmentation IOU to evaluate the accuracy of the semantic map generated by the proposed method. The calculated segmentation IOU was 0.5667. If the segmentation IOU is above 0.5, it means that more than two-thirds of the predicted semantic map overlap ground truth.

The result shows that the proposed real-to-sim method can improve the quality of sonar images by removing the degradation effect and generate an accurate semantic map of the underwater scene. The semantic map can provide accurate information of underwater terrain through the size and location of highlights and shadows. Therefore, the proposed method can be utilized as a pre-processing method of sonar images to extract more reliable information and improve the precision of sonar-image-based operation.

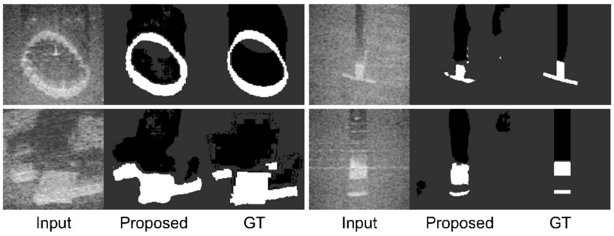


Fig. 10. Results of the semantic map generation using the proposed method.

4.3. Object detection test in field

We applied the proposed realistic sonar image generation method to underwater object detection. We trained the YOLO network by generating realistic images of the target object using the proposed method and checked whether the trained network could accurately detect the target object in the sea trials. A DNN tends to fit into the domain of training data as training progresses. Therefore, if the proposed method does not generate realistic images, the trained network will fail to detect the target objects in the real sonar images of the field. By training the CNN with only the generated realistic images and testing the CNN with the real sonar images, we can verify that the proposed method generates realistic images that are similar enough to the field data.

We first generated the training dataset for the YOLO network. We set the target object to detect as a tire, which is one of the frequently found marine waste offshore near ports. We created a 3D model of the tire like Fig. 11. Then, the ray-tracing-based simulation and sim-to-real translation generated the sonar images of the objects. Since accurate object detection requires training images of the target objects in various conditions, we rotated and translated the objects, and 36 source images are generated. We then resized the source images to a predefined size and placed them in a random position on a 512×96 canvas to diversify the training data. Black pixels are padded in the upper and lower areas of the canvas outside the images, considering the shadow region where no echoes are formed in the real sonar image. As a result, we generated a training data set consisting of 108 images like Fig. 12. We then trained the YOLO network for 2,400 epochs with these generated images. The training took about an hour using the GPU Titan V.

Field experiments were conducted to test the trained network. We installed a tire on the seabed of Janggil-bay, Pohang, Korea. In addition to tires, other shapes of objects were also installed to verify whether the proposed method can distinguish between target and other objects. Then, we captured the images of the underwater scene using the DIDSON while moving in the nearby sea. Then,

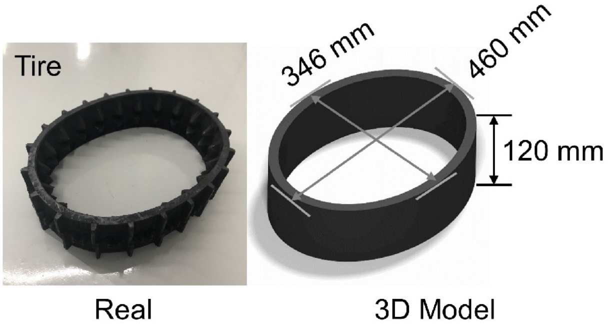


Fig. 11. Detection target (tire) and its modeled 3D shape.

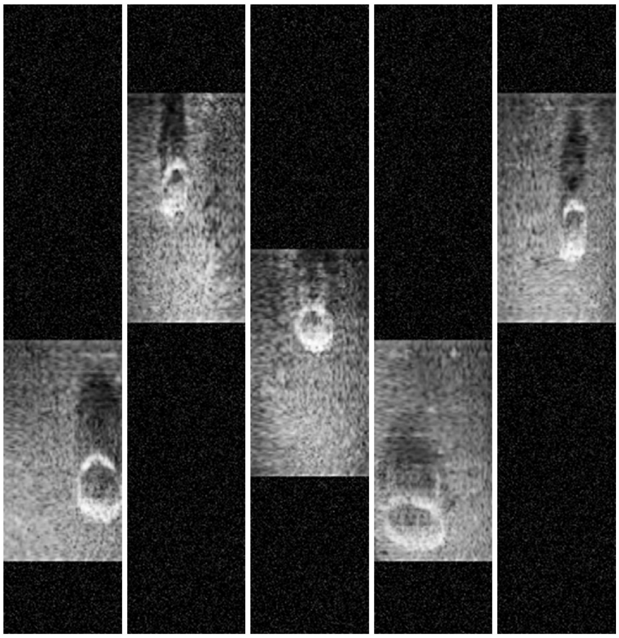


Fig. 12. Images of tire generated by the proposed method to train the YOLO network.

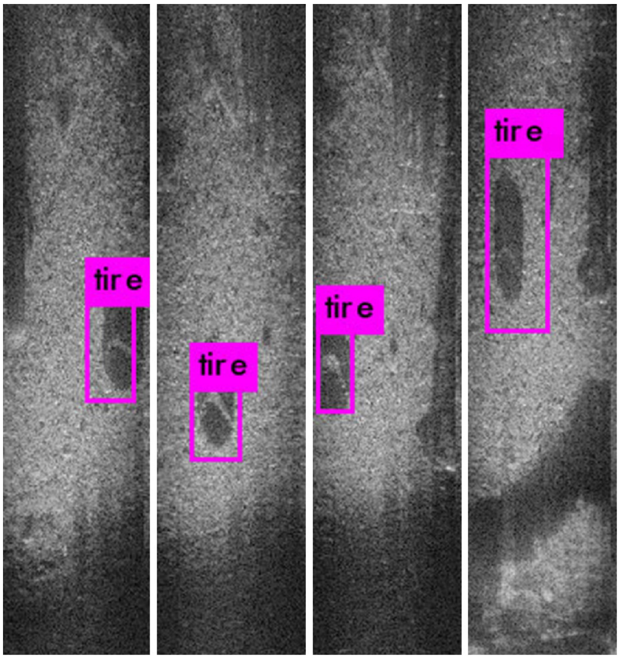


Fig. 13. Target object (tire) detection results using the proposed method.

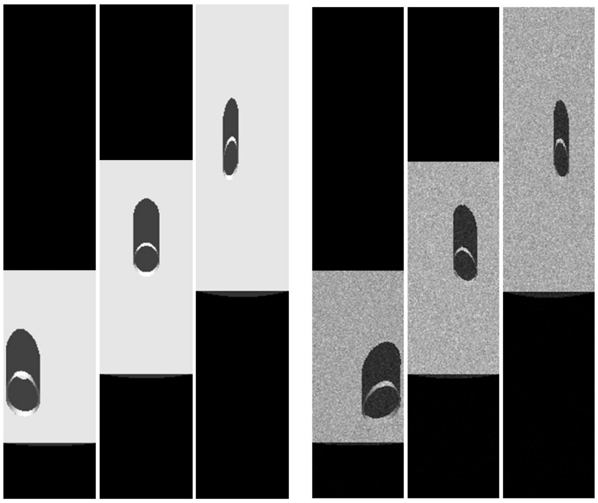
we checked the trained network can detect the target object in the captured sonar images. For the test, a total of 127 images were acquired through the field experiments.

The network trained with the proposed method could accurately detect the target object in the sonar images captured at sea, like Fig. 13. The trained network successfully detected the objects placed in various locations. The network could also detect the object well even if only a part of the object was taken. Moreover, if there were other objects which also have highlights and shadows, the trained network could detect only the object correctly.

For the quantitative evaluation, we measured the detection accuracy. In the images of the test dataset, there are 67 target objects and 75 non-target objects. We counted how many of the target objects were accurately detected and how many of the non-target objects were not erroneously detected. As a result, the network recorded a true positive rate of 86.6 % and a true negative rate of 88.0 %.

We also measured the processing speed of the proposed object detection method. In underwater exploration, the operational time is limited, and the reproducibility of the exploration is not high, so the fast processing speed is essential for the algorithms. The proposed method recorded a processing speed of 37.08 frames per second (fps). The sonar sensor we used has a frame rate of 4-21 fps. So, the proposed method can detect the target objects in real-time using the sonar sensor.

To prove the effectiveness of the sim-to-real translation in generating realistic sonar images, we compared the object detection results with the other two YOLO models. One model was trained with images simulated by the simple ray-tracing method like Fig. 14(a), and another one was trained with images generated by adding speckle noise to the simulated image like Fig. 14(b). Those two models failed to detect the target object in real sonar images with less than 10 % detection accuracy. As a result, we confirmed that the proposed method using the ray tracing and GAN-based sim-to-real translation could generate realistic sonar images effectively.



(a) Simulated by the ray tracing (b) Ray tracing + Speckle noise

Fig. 14. Images simulated to train other networks for comparison.

Through application to object detection, we verified that the proposed method could generate realistic images that can be utilized in other algorithms. As shown in the object detection results, although the YOLO network was trained only with the generated images without the images captured at the field, the network can detect the target objects even in images captured at sea. It presents that the proposed method can generate realistic images that have similar features to the images taken at sea. Therefore, the proposed method can be applied to more various sonar-image-based algorithms to make the development more efficient and to improve the robustness.

5. CONCLUSION

In this paper, we proposed a method to synthesize realistic sonar images. The method used a ray-tracing algorithm to calculate semantic information of a viewed scene. Then, the GAN generates realistic sonar images preserving essential information. The GAN can also generate a semantic map of a viewed scene from a real sonar image by segmentation. The generated realistic sonar images were successfully utilized as training images for object detection in the field experiments. The semantic map would also be helpful to extract more reliable information from the real sonar images. Therefore, the proposed method can be useful for the data augmentation and pre-processing of sonar images.

REFERENCES

[1] H.-S. Kim and D. Lee, "Intelligent psr estimator for feature extraction of a passive sonar target," *International Journal of Control, Automation and Systems*, vol. 8, no. 3, pp. 677-682, 2010.

[2] T. Kim, J. Kim, and S.-W. Byun, "A comparison of nonlinear filter algorithms for terrain-referenced underwater navigation," *International Journal of Control, Automation and Systems*, vol. 16, no. 6, pp. 2977-2989, 2018.

[3] M. B. Loc, H.-S. Choi, J.-M. Seo, S.-H. Baek, and J.-Y. Kim, "Development and control of a new auv platform," *International Journal of Control, Automation and Systems*, vol. 12, no. 4, pp. 886-894, 2014.

[4] M. G. Joo and Z. Qu, "An autonomous underwater vehicle as an underwater glider and its depth control," *International Journal of Control, Automation and Systems*, vol. 13, no. 5, pp. 1212-1220, 2015.

[5] M. H. Lee, J.-H. Moon, I.-S. Kim, C. S. Kim, and J. W. Choi, "Pre-processing faded measurements for bearing-and-frequency target motion analysis," *International Journal of Control, Automation, and Systems*, vol. 6, no. 3, pp. 424-433, 2008.

[6] J. Kim and S.-C. Yu, "Convolutional neural network-based real-time rov detection using forward-looking sonar image," *Proc. of IEEE/OES Autonomous Underwater Vehicles (AUV)*, IEEE, pp. 396-400, 2016.

[7] D. P. Williams, "Underwater target classification in synthetic aperture sonar imagery using deep convolutional neural networks," *Proc. of 23rd international conference on pattern recognition (ICPR)*, IEEE, pp. 2497-2502, 2016.

[8] H. Horimoto, M. Toshihiro, K. Kofuji, and T. Ishihara, "Autonomous sea turtle detection using multi-beam imaging sonar: Toward autonomous tracking," in *2018 IEEE/OES Autonomous Underwater Vehicle Workshop (AUV)*, IEEE, pp. 1-4, 2018.

[9] J. Kim, M. Sung, and S.-C. Yu, "Development of simulator for autonomous underwater vehicles utilizing underwater acoustic and optical sensing emulators," *Proc. of 18th International Conference on Control, Automation and Systems (ICCAS)*, IEEE, pp. 416-419, 2018.

[10] S. Kwak, Y. Ji, A. Yamashita, and H. Asama, "Development of acoustic camera-imaging simulator based on novel model," *Proc. of IEEE 15th International Conference on Environment and Electrical Engineering (EEEIC)*, IEEE, pp. 1719-1724, 2015.

[11] J. Riordan, F. Flannery, D. Toal, M. Rossi, and G. Dooly, "Interdisciplinary methodology to extend technology readiness levels in sonar simulation from laboratory validation to hydrography demonstrator," *Journal of Marine Science and Engineering*, vol. 7, no. 5, p. 159, 2019.

[12] R. Cerqueira, T. Trocoli, G. Neves, S. Joyeux, J. Albiez, and L. Oliveira, "A novel gpu-based sonar simulator for real-time applications," *Computers & Graphics*, vol. 68, pp. 66-76, 2017.

[13] H. Joe, J. Kim, and S.-C. Yu, "Sensor fusion-based 3d reconstruction by two sonar devices for seabed mapping," *IFAC-PapersOnLine*, vol. 52, no. 21, pp. 169-174, 2019.

[14] B. Kim, J. Kim, M. Lee, M. Sung, and S.-C. Yu, "Active planning of auvs for 3d reconstruction of underwater object using imaging sonar," *Proc. of IEEE/OES Autonomous Underwater Vehicle Workshop (AUV)*, IEEE, pp. 1-6, 2018.

[15] P. C. Etter, "Underwater acoustic modeling: principles, techniques and applications," 1995.

[16] M. Palmese and A. Trucco, "Acoustic imaging of underwater embedded objects: Signal simulation for three-dimensional sonar instrumentation," *IEEE transactions on instrumentation and measurement*, vol. 55, no. 4, pp. 1339-1347, 2006.

[17] B. K. Novikov, O. V. Rudenko, and V. I. Timoshenko, *Nonlinear Underwater Acoustics*, American Institute of Physics, New York, 1987.

[18] B. Kim and S.-C. Yu, "Imaging sonar based real-time underwater object detection utilizing adaboost method," in *2017 IEEE Underwater Technology (UT)*, IEEE, pp. 1-5, 2017.

[19] C. H. Yu and J. W. Choi, "Interacting multiple model filter-based distributed target tracking algorithm in underwater wireless sensor networks," *International Journal of Control, Automation and Systems*, vol. 12, no. 3, pp. 618-627, 2014.

[20] D. Kim, J.-U. Shin, H. Kim, H. Kim, D. Lee, S.-M. Lee, and H. Myung, "Development and experimental testing of an autonomous jellyfish detection and removal robot system," *International Journal of Control, Automation and Systems*, vol. 14, no. 1, pp. 312-322, 2016.

[21] M. Sualeh and G.-W. Kim, "Simultaneous localization and mapping in the epoch of semantics: A survey," *International Journal of Control, Automation and Systems*, vol. 17, no. 3, pp. 729-742, 2019.

[22] S. Lee, B. Park, and A. Kim, "Deep learning from shallow dives: Sonar image generation and training for underwater object detection," *arXiv preprint arXiv:1810.07990*, 2018.

[23] J. L. Chen and J. E. Summers, "Deep neural networks for learning classification features and generative models from synthetic aperture sonar big data," *Proceedings of Meetings on Acoustics 172ASA*, vol. 29, no. 1, ASA, p. 032001, 2016.

[24] A. S. Glassner, *An Introduction to Ray Tracing*, Elsevier, 1989.

[25] E. Catmull, "A subdivision algorithm for computer display of curved surfaces," Utah Univ. Salt Lake City School of Computing, Tech. Rep., 1974.

[26] P. Isola, J.-Y. Zhu, T. Zhou, and A. A. Efros, "Image-to-image translation with conditional adversarial networks," *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1125-1134, 2017.

[27] O. Ronneberger, P. Fischer, and T. Brox, "U-net: convolutional networks for biomedical image segmentation," *Proc. of International Conference on Medical Image Computing and Computer-assisted Intervention*, Springer, pp. 234-241, 2015.

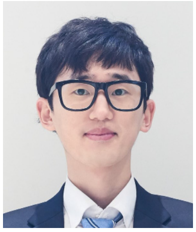
[28] J. Moon, H. Kim, and B. Lee, "View-point invariant 3d classification for mobile robots using a convolutional neural network," *International Journal of Control, Automation and Systems*, vol. 16, no. 6, pp. 2888-2895, 2018.

[29] S.-H. Kim and H.-L. Choi, "Convolutional neural network for monocular vision-based multi-target tracking," *International Journal of Control, Automation and Systems*, vol. 17, no. 9, pp. 2284-2296, 2019.

[30] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You only look once: Unified, real-time object detection," *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 779-788, 2016.

[31] E. Belcher, W. Hanot, and J. Burch, "Dual-frequency identification sonar (didson)," in *Proceedings of the 2002 Interntional Symposium on Underwater Technology (Cat. No. 02EX556)*, IEEE, pp. 187-192, 2002.

[32] Z. Wang, A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli, "Image quality assessment: from error visibility to structural similarity," *IEEE Transactions on Image Processing*, vol. 13, no. 4, pp. 600-612, 2004.



Minsung Sung received his B.E. degree from the Pohang University of Science and Technology (POSTECH), Pohang, Korea, in 2016, where he is currently pursuing a Ph.D. degree with the Department of IT Engineering. He is a member of the Hazardous and Extreme Environment Robotics (HERO) Laboratory, POSTECH. His research interests include deep learning, sonar image processing, underwater perception with sensor fusion, and SLAM.

Jason Kim received his B.E. degree in computer engineering in 2018 from the Ulsan National Institute of Science and Technology (UNIST), Ulsan, Korea. Currently, he is an M.S. candidate in the Department of IT Engineering at Pohang University of Science and Technology (POSTECH), Pohang, Korea. His research interests include computer simulation, underwater optical/sonar image processing, and SLAM.

Meungsuk Lee received his B.E. degree in mechanical engineering in 2018 from Pohang University of Science and Technology (POSTECH), Pohang, Korea, where he is currently pursuing an M.S. degree with the Department of Electrical Engineering. His research interests include underwater robotics and underwater vision.

Byeongjin Kim received his B.E. degree in electrical engineering from the Pohang University of Science and Technology (POSTECH), Pohang, Korea, in 2015, where he is currently pursuing a Ph.D. degree with the Department of IT Engineering. His research interests include underwater robotics, sonar image processing, and SLAM.

Taesik Kim received his B.E. degree in mechanical engineering from the Ulsan National Institute of Science and Technology (UNIST), Ulsan, Korea. He is currently pursuing a Ph.D. degree with the Department of IT Engineering, Pohang University of Science and Technology (POSTECH), Pohang, Korea. His research interests include underwater robotics, hydrodynamics, and manipulation.

Juhwan Kim received his B.E. degree in electrical engineering in 2015 from the Pohang University of Science and Technology (POSTECH), Pohang, Korea, where he is currently pursuing a Ph.D. degree with the Department of IT Engineering. His research interests include underwater robotics, machine learning, and multi-agent underwater manipulation.

Son-Cheol Yu received his M.E. and Ph.D. degrees from the Department of Ocean and Environmental Engineering, University of Tokyo, in 2000 and 2003, respectively. He is an Associate Professor of the Department of IT Engineering, Electrical Engineering, and Advanced Nuclear Engineering with the Pohang University of Science and Technology (POSTECH), Korea. He is also the Director of Hazardous and Extreme Environment Robotics (HERO) Lab, IEEE Ocean Engineering Society Korea Chapter, Gyeongbuk Sea Grant Center. He has been a Researcher of mechanical engineering with the University of Hawaii from 2004 to 2007 and an Assistant Professor of mechanical engineering with the Pusan National University from 2008 to 2009. His research interest is autonomous underwater Vehicles, underwater sensing, and multi-agent-based robotics.

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.