Development of a Side Scan Sonar Module for the UnderWater Simulator

Dae-Hyeon Gwon[1], Joowan Kim[2], Moon Hwan Kim[3], Ho Gyu Park[4], Tae Yeong Kim[5], Ayoung Kim*
[1,2,*]Dept. of Civil and Environmental Engineering, Korea Advanced Institute of Science and Technology, Daejeon, 34141, Korea
(Tel : +82-42-350-3672; E-mail: [kdhx13,jw_kim,ayoungk]@kaist.ac.kr)
[3,4,5]LIG Nex1 Maritime Research Center, Seongnam-si, Gyeonggi-do, 13488, Korea
(Tel : +82-31-8026-7000; E-mail: [moonanikim,hgpark77,taeyeong.kim]@lignex1.com)

***Abstract*—In this paper, we implemented the Side Scan Sonar (SSS) module for the UnderWater Simulator (UWSim) [1], specifically focusing on sonar image simulation. To obtain the simulated SSS images, we adopted the Lambertian diffusion model [2]. The beam profile was modeled using a pair of sine wave functions. We developed this sonar module in the UWSim which runs on Robot Operating System (ROS) framework. For validation, we applied feature matching to the obtained sonar images, and examined the inliers and matching performance.**

***Keywords*—Side Scan Sonar, UnderWater Simulator, AUV, Image simulation**

1. Introduction

Acquiring real-world data is challenging, especially for underwater robotics engineers. Toward this issue, exploiting simulators shed light on the data shortage by providing simulated data and the mean to validate the algorithm. The simulators are usually equipped with various underwater sensors for navigation and perception. We focus on one of the popular underwater simulators, UWSim, developed by IRSLab of Jaume-I University of Castelln [1]. The simulator provides a variety of underwater sensors to fully simulate the environment, including inertial measurement unit (IMU). Doppler velocity log (DVL), and cameras.

In this paper, we select Side Scan Sonar (SSS) as an imaging sensor. Although cameras are commonly used for imaging, the camera image deteriorates by the high speckle noise of the underwater. Compared to these limited vision sensors, sonar wave transmits well underwater, and sonar is widely used for underwater imaging [3]. In this paper, we choose the SSS among several sonar types. The SSS which is used in this work provides a high resolution of 10 $\mathrm{cm} \times 5\ \mathrm{cm}$ with a wide sensing range of 70 m [4].

We developed a SSS module in UWSim [1]. Unlike the MALAB based bathymeter simulator [5], the proposed SSS module was developed in C++ language which provided faster computing performance than MATLAB. This proposed SSS module improves UWSim by adding an SSS module. We implemented a black-brown-yellow colormap, Rayleigh noise [6] and Speckle noise [7] which were intended to approximate an artificial SSS image to a real image.

We have tested A-KAZE feature matching [8] to the simulated SSS images for validation. Among many visual features, A-KAZE was selected because of its reported outperformance and fast computation time [8].

2. Related Research

Underwater simulators are essential for underwater engineers as enabling researchers to overcome insufficient data that is often encountered during underwater experiments. Underwater simulators have been previously implemented by researchers. For example, UWSim [1] works with the open-source Ubuntu ROS. It uses Open Scene Graph (OSG) which includes robot dynamics for underwater robots as well as fluid dynamics for underwater environments. OSG has the ability of displaying an image stream of underwater robots in real time with high graphic quality. In the study by Heriot-Watt University OSL [5], authors presented sonar sidescan simulator and bathymeter simulator using Matlab. Their work only focused on Matlab without the ROS interface, whereas our work targets developing a missing sonar module for the UWSim in the ROS framework.

The fundamentals of SSS were handled by Helferty [9], such as physical properties, data processing, image composition and enhancement. The simulation of the SSS image performed by Vanessa creates the SSS image through ray tracing rendering technology [10]. A more detailed SSS image model was introduced in [11]. Later, a simplified SSS image model was presented by using the Lambertian diffusion model, and has been widely used in sonar image modeling by many researchers [2], [4], [5], [12]. In this work, the SSS beam profile was approximated to a plane to reduce the computational cost from three-dimensional (3D) to two-dimensional (2D). The beam waveform of the SSS was computed through experiments or approximation since the theoretical expression is hardly obtained. In this work, we approximated the beam waveform by a pair of sinusoidal functions similar to the work by [13]. In the field of underwater sonar imaging, there are typical low and high frequency noise of the underwater environment. In this paper, the low frequency noise was modeled by Rayleigh noise [6], and the high frequency noise was modeled by Speckle noise [7].

SSS image matching using a correlation of image intensities and using wavelet transform for denoising were proposed by [14]. The state of the art features were compared to match

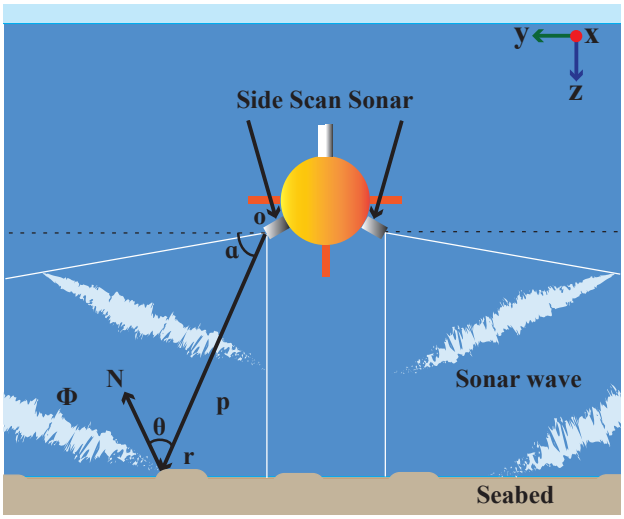


Fig. 1. Side scan sonar coordinate

TABLE 1
SIDE SCAN SONAR PARAMETERS

Symbol	explanation
o	Origin point of sensor coordinate
p	Reflection point of seabed
$\mathbf{r}$	Vector of points o and p
$\mathbf{N}$	Normal vector at p
θ	Angle between $\mathbf{r}$ and $\mathbf{N}$
α	Angle between x-axis and $\mathbf{r}$
Φ	Side scan sonar beam profile
R	Reflection coefficient
$I(p)$	Image intensity of sonar wave
K	$1/I_{max}$

the SSS images [15]. For example, Scale Invariant Feature Transform (SIFT) [16], Speeded Up Robust Features (SURF) [17], Oriented FAST and Rotated BRIEF (ORB) [18] and A-KAZE [8] were compared. As reported by [8], inlier ratio for the A-KAZE for SSS images overwrote other feature extractors. Therefore, we also test A-KAZE feature detector to the simulated sonar images.

3. SIDE SCAN SONAR MODULE DEVELOPMENT

3.1. Side Scan Sonar Modeling

We adopted the simplified model by [12]. The SSS model was approximated using the Lambertian diffusion model, using the coordinate system shown in the following figure.

The model (1) is a simplified Lambertian diffusion model which is applied to sound waves [12].

$$\begin{aligned} I(p) &= RK\Phi\cos(\theta) \\ \cos(\theta) &= \frac{\mathbf{r}\cdot\mathbf{N}}{\|\mathbf{r}\|\,\|\mathbf{N}\|} \end{aligned} \tag{1}$$

The parameters for the equation and Fig. 1 are summarized in Table 1. Since the reflection coefficient R is a function of the parameter at the point p and the sonar frequency, and the reflection coefficient R was assumed to be a constant as the frequency is constant at the sand bottom. In the simulation, $R = 0.5$ was used. Then, o and p were calculated from the data obtained from the sensor and K was calculated. While it is essential to evaluate sonar waveforms, only waveform data is within the range of possibility to be obtained experimentally without formulas. Therefore, it is available to simulate by assuming approximate expressions [13].

3.2. Side Scan Sonar Module in ROS

The data flow between UWSim and ROS illustrates in Fig. 2. The UWSim corresponds with ROS through ROS interface. In ROS interface, the data of UWSim converts into ROS topic which is published into ROS. We developed SSS module by modifying the ROS interface. The Lambertian diffusion model [12] was implemented in the ROS interface.

From (1), we obtain a grayscale image which is the raw data of SSS images. This image is the SSS image topic in Fig. 2. The noise is added to make the synthetic image similar to the actual image such as Rayleigh noise for low frequency noise [6] and Speckle noise for high frequency [7] which is the physical nature of underwater.

These noises have been used for sidescan simulator by [5]. Rayleigh noise was developed by adding Rayleigh distribution to image intensity, and Speckle noise was exploited by multiplying uniform noise by image intensity. Each noise formula is as follows.

$$\begin{aligned} I' &= I\cdot(1+u)+r \\ u &\sim U(0,\gamma) \\ r &\sim R(0,\sigma) \end{aligned} \tag{2}$$

The I is image intensity, I' is image intensity with noise, u is uniform noise U with variance γ and r is Rayleigh noise R with variance σ. Therefore, the output image has both Rayleigh noise for low frequency noise and Speckle noise for high frequency.

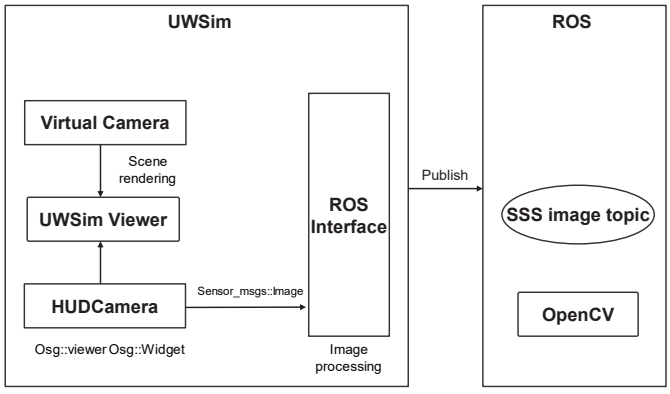


Fig. 2. Data flow between UWSim and ROS

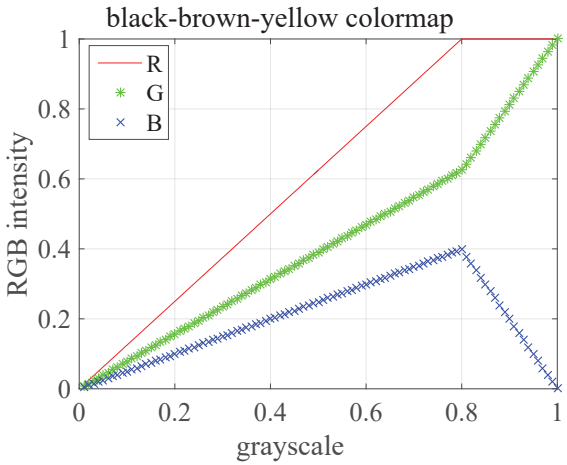


Fig. 3. The black-brown-yellow colormap for simulated images

Now we introduce our colormap for SSS image simulation. The colormap was applied in ROS with OpenCV. When the size of the grayscale image is 0 to 1, we defined the RGB channel of the colormap as (3).

$$
\begin{aligned}
R(I) &= \begin{cases} 1.25I, & (0 \le I < 0.8) \\ 1, & (0.8 \le I \le 1) \end{cases} \\
G(I) &= \begin{cases} 0.7812I, & (0 \le I < 0.8) \\ 1.883I - 0.883, & (0.8 \le I \le 1) \end{cases} \\
B(I) &= \begin{cases} 0.4975I, & (0 \le I < 0.8) \\ -1.99I + 1.99, & (0.8 \le I \le 1) \end{cases}
\end{aligned} \tag{3}
$$

This colormap is shown in Fig. 3. The RGB channel was $(R, G, B) = (0, 0, 0)$ when $I = 0$ so to be converted to black. On the other hand, the RGB channel was $(1, 0, 625, 0.398)$ when the grayscale $I = 0.8$ in order to be mapped to brown. The RGB channel was $(1, 1, 0)$ when grayscale $I = 1$ because we made it to convert to yellow. In the middle, the colormap was constructed by connecting linearly as shown in Fig. 3.

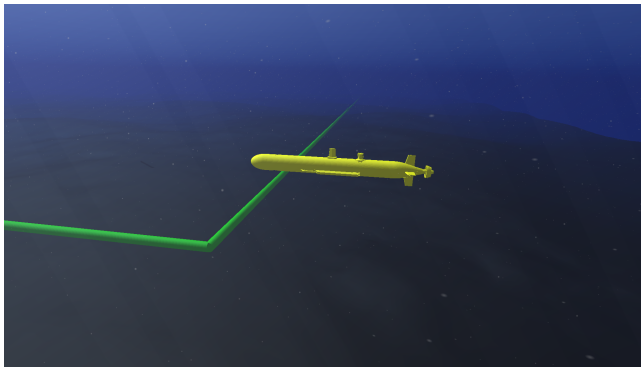
4. Simulation

4.1. Test Environment in UWSim

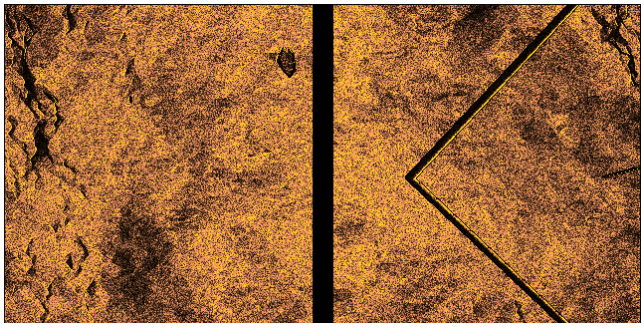
Using this simulated sonar image, we tested the feature matching algorithm for validation. Fig. 4(a) illustrates the simulation pipe inspection scenario in underwater environments. The underwater pipes were 480m long each, and the four pipes were placed on the sea floor to form a square. In addition, there were four rocks of 1.7 m length on each of the four corners. Fig. 4(b) illustrates the developed SSS module.

4.2. Feature Matching Comparison

Within this simulated environment, we selected A-KAZE for SSS image matching. The Fig. 5(a) shows the correspondences of images by A-KAZE. The Table 2 summaries the matching performance of feature extractors among most widely used features in the computer vision area. For instance,



(a) View of simulation environment with UWSim

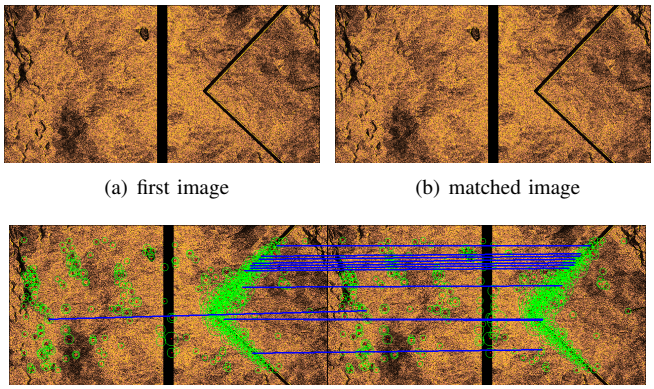


(b) SSS image sample

Fig. 4. View of the UWSim and developed SSS module.

SIFT [16], SURF [17], ORB [18] and A-KAZE [8] were compared with number of inliers. These feature extractors compared the features between two images. Then, Nearest Neighbor Distance Ratio (NNDR) matching was performed to find the matching points of these features. Finally, the random sample consensus (RANSAC) was applied to remove outliers from matching points and left only inliers from each images. The number of inliers were 8, 7, 4, and, 11 for SIFT [16], SURF [17], ORB [18] and A-KAZE [8] respectively. As a result, A-KAZE [8] had the largest number of inliers. Although we had not tested with parameter optimization, we believe A-KAZE [8] parameter optimization can further improve the inlier ratio.

Since the underwater sonar images contain high speckle noise and have low resolution, many feature matching methods may not successful for these sonar images. To tackle this issue, image preprocessing is essential. Since we added Rayleigh noise [6] and Speckle noise [7] to the SSS image, it is needed preprocessing such as image enhancement and noise reduction. We believe a proper preprocessing would improve the feature matching performance even for side scanning sonar images. Potential candidates are median filter or bilateral filter. The median filter is widely selected to noise filtering, and the bilateral filtering preserves edges of image alleviating the noise. Because edges are expect to become feature points, the bilateral filtering may provide better feature matching performance.



(a) first image (b) matched image

(c) A-KAZE image matching

Fig. 5. (a) The first image was obtained from the starting point. (b) The matched image was extracted from the loop closing point. (c) Image matching pairs from SSS image matching using A-KAZE.

TABLE 2
FEATURE MATCHING COMPARISON

	Number of Inliers
SIFT	8
BRISK	7
ORB	4
A-KAZE	11

5. CONCLUSION

We developed the SSS module for UWSim [1]. As a preliminary step to developing a SSS simulator operated on UWSim [1], we acquired a simulated SSS image using the Lambertian diffusion model [12]. The simplified SSS image equation was used, and the beam profile was estimated based on the sinusoidal waveform. For validation, we used A-KAZE [8] feature matching to the simulated images. We implemented a black-brown-yellow colormap and applied Rayleigh noise [6] and Speckle noise [7] to reflect the physical characteristic of underwater noise.

For the future work, we plan to use A-KAZE [8] feature matching for SSS bundle adjustment (BA). Robot odometry and SSS images will generate for pose-graph simultaneous localization and mapping (SLAM). The output of SSS BA will be then merged to SLAM framework for loop closing constraints of pose-graph SLAM.

ACKNOWLEDGEMENTS

This work is supported by LIG Nex1. Joowan Kim was financially supported by Korea Ministry of Land, Infrastructure and Transport (MOLIT) via 'U-City Master and Doctor Course Grant Program'.

REFERENCES

[1] M. Prats, J. Prez, J. J. Fernndez, and P. J. Sanz, "An open source tool for simulation and supervision of underwater intervention missions," in *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, Oct 2012, pp. 2577–2582.

[2] H. Ragheb and E. R. Hancock, "Surface radiance correction for shape from shading," *Proceedings of the International Conference Pattern Recognition*, vol. 38, no. 10, pp. 1574 – 1595, 2005.

[3] M. F. Fallon, M. Kaess, H. Johannsson, and J. J. Leonard, "Efficient auv navigation fusing acoustic ranging and side-scan sonar," in *Proceedings of the IEEE International Conference on Robotics and Automation*, May 2011, pp. 2398–2405.

[4] D. Langer and M. Hebert, "Building qualitative elevation maps from side scan sonar data for autonomous underwater navigation," in *Proceedings of the IEEE International Conference on Robotics and Automation*, Apr 1991, pp. 2478–2483 vol.3.

[5] Y. Pailhas, Y. Petillot, C. Capus, and K. Brown, "Real-time sidescan simulator and applications," in *Proceedings of the IEEE OCEANS-Europe Conference and Exhibition*, May 2009, pp. 1–6.

[6] A. S. A. Ghani and N. A. M. Isa, "Underwater image quality enhancement through rayleigh-stretching and averaging image planes," *International Journal of Naval Architecture and Ocean Engineering*, vol. 6, no. 4, pp. 840–866, 2014.

[7] S. Banerjee, R. Ray, S. N. Shome, and G. Sanyal, "Noise induced feature enhancement and object segmentation of forward looking sonar image," *Procedia Technology*, vol. 14, pp. 125–132, 2014.

[8] P. F. Alcantarilla, A. Bartoli, and A. J. Davison, "KAZE features," in *Proceedings of the European Conference on Computer Vision*, 2012.

[9] M. Helferty, "The geological interpretation of side-scan sonar," *Reviews of Geophysics*, vol. 28, pp. 357–380, 1990.

[10] V. S. Blake, "Simulation in underwater archaeological prospection," *Conference of the Remote Sensing society.*, 1996.

[11] S. Anstee, "Removal of range-dependent artifacts from sidescan sonar imagery," DTIC Document, Tech. Rep., 2001.

[12] E. Coiras, Y. Petillot, and D. M. Lane, "Multiresolution 3-d reconstruction from side-scan sonar images," *IEEE Transactions on Image Processing*, vol. 16, no. 2, pp. 382–390, Feb 2007.

[13] C. de Jong, G. Lachapelle, S. Skone, and I. Elema, "Multibeam sonar theory of operation," Delft University Press Delft, the Netherlands, Tech. Rep., 2002.

[14] N. Neretti, N. Intrator, and Q. Huynh, "Target detection in side-scan sonar images: expert fusion reduces false alarms," 2002.

[15] X.-F. Ye, P. Li, J.-G. Zhang, J. Shi, and S.-X. Guo, "A feature-matching method for side-scan sonar images based on nonlinear scale space," *Journal of Marine Science and Technology*, vol. 21, no. 1, pp. 38–47, 2016.

[16] D. Lowe, "Distinctive image features from scale-invariant keypoints," *International Journal of Computer Vision*, vol. 60, no. 2, pp. 91–110, 2004.

[17] H. Bay, A. Ess, T. Tuytelaars, and L. Van Gool, "Speeded-up robust features (SURF)," *Computer Vision and Image Understanding*, vol. 110, no. 3, pp. 346–359, 2008.

[18] E. Rublee, V. Rabaud, K. Konolige, and G. Bradski, "Orb: An efficient alternative to sift or surf," in *Proceedings of the IEEE International Conference on Computer Vision*, Nov 2011, pp. 2564–2571.