ALAR: Customizable Multimodal Underwater Scenarios for Harsh-Domain Perception

Andrea Bedei[1], Lorenzo Bacchiani[1], Nicholas Petrucci[2],
Andrea Melis[1], Franco Callegati[1], Giovanni Pau[1], Roberto Casadei[1], Roberto Girau[1]
[1] *Department of Computer Science and Engineering, University of Bologna, Bologna, Italy*
[2] *Department of Industrial Design, University of Bologna, Bologna, Italy*
{andrea.bedei3, lorenzo.bacchiani2, nicholas.petrucci2, a.melis,
franco.callegati, giovanni.pau, roby.casadei, roberto.girau}@unibo.it

***Abstract*—Perception in harsh underwater conditions remains constrained by turbidity, low illumination, sonar artifacts, and the scarcity of multimodal underwater datasets. Existing synthetic solutions help address this data scarcity, but they remain limited when flexible scene regeneration and acoustically coherent multimodal acquisition are required. In particular, runtime scene changes in modern underwater simulators, e.g., HoloOcean, are not natively reflected with reliable acoustic consistency, preventing straightforward procedural generation of sonar-aware multimodal data. We present *ALAR*, a runtime-configurable extension of HoloOcean for multimodal underwater dataset generation. Our extension introduces runtime scene-population API for custom assets with an explicit sonar refresh mechanism, ensuring that inserted geometry remains acoustically visible. Using *ALAR*, we generated a dataset of 27022 synchronized multimodal samples over 292 runs, combining front and bottom RGB imagery, sonar data, and rover-state metadata. To evaluate the usability of our extension for real-world applications, we train the ResNet18 neural network over synthetic data and validate it on the UXO dataset, which provides real sonar torpedo images. Results show that the neural network reaches 73.33% torpedo recall, proving that *ALAR* is a practical basis for real multimodal underwater perception.**

***Index Terms*—Underwater perception, synthetic dataset, imaging sonar, multimodal simulation, harsh domains, sim-to-real**

I. Introduction

Autonomous perception in underwater environments is fundamentally constrained by turbidity, light attenuation, sparse visual texture, and acoustic artifacts, which collectively degrade both optical and sonar-based modalities in ways that have no direct analogue in terrestrial robotics [1], [2]. Imaging sonar remains essential precisely because it retains geometric discriminability when optical sensing collapses [3], yet the research community's ability to develop and evaluate learning-based sonar perception is severely curtailed by a persistent data scarcity problem [4], [5]. Real multimodal datasets do exist, e.g., the UXO dataset [6] is a notable example, but they remain limited in number and are typically task-specific, modality-incomplete, or otherwise too narrow for systematic sim-to-real experimentation [7]. Synthetic corpora [8] help fill the training gap, but still leave two practical needs unmet.

The first is *runtime reconfigurability*: the ability to insert assets, vary target and clutter layouts, and regenerate scenes programmatically without repeated manual editing of the simulator. The second is *acoustic consistency*: the guarantee that each regenerated scene yields physically coherent sonar observations where runtime-inserted objects are correctly represented by the acoustic sensor. These two needs are coupled: runtime scene adaptation is only useful for dataset generation if the sonar reflects each updated scene state reliably.

Modern simulators, e.g., HoloOcean [9], [10], provide realistic underwater physics and sonar simulation, making it a strong foundation for dataset generation. However, when used as a procedural acquisition engine rather than a static scene simulator, a concrete obstacle arises: sonar-side cached geometric structures can persist after runtime scene changes, causing newly inserted objects to be missing or inconsistently represented in the acoustic view unless the relevant geometry is explicitly invalidated and rebuilt.

To address this gap, we introduce *ALAR*, a runtime-configurable extension of HoloOcean for multimodal underwater dataset generation. As summarized in fig. 1, *ALAR* takes vehicle and sensor configurations, an asset library, and a scenario configuration as inputs (blue blocks), and produces synchronized multimodal dataset samples, ground-truth annotations, and test environments as outputs (orange blocks). After each scene update, *ALAR* executes an explicit sonar refresh stage that invalidates and rebuilds the sonar-side cached geometry, ensuring that runtime-inserted objects remain acoustically observable. Using this pipeline, we generate a multimodal dataset of 27022 synchronized samples and evaluate its practical utility through a zero-shot sim-to-real transfer experiment on real UXO sonar imagery, using ResNet18 as a lightweight and widely adopted baseline for constrained deployment scenarios.

The main contributions of this paper are:

- *ALAR*, an extension of HoloOcean supporting runtime insertion of custom underwater assets with explicit sonar cache management after scene changes, enabling acoustically coherent multimodal acquisition across arbitrary target configurations. *ALAR* and the custom assets used in this work are publicly available at [11];
- a publicly available multimodal dataset [12] of 27022 synchronized samples generated with *ALAR*, where each sample includes front and bottom RGB imagery, imaging sonar data, a derived polar sonar image, and rover-state metadata;

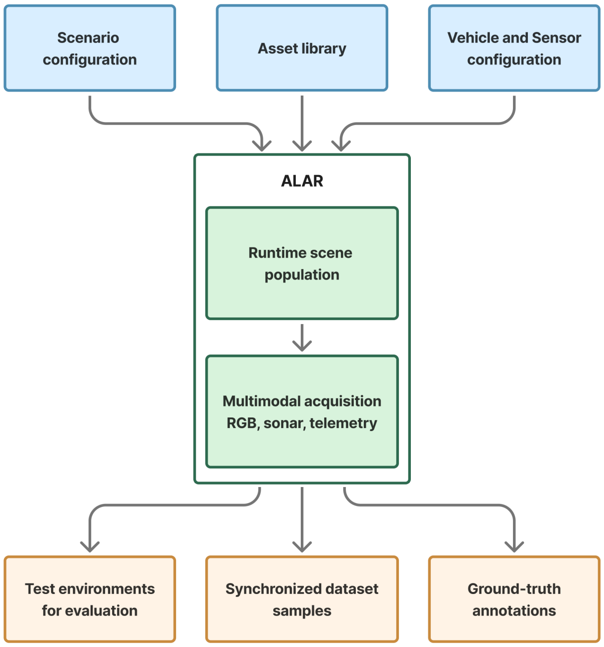

Fig. 1. High-level overview of *ALAR* workflow.

- a zero-shot sim-to-real transfer benchmark where a ResNet18 classifier is trained on synthetic data and tested over the UXO dataset.

The remainder of the paper is organized as follows. Section II reviews related work. Section III describes the simulator extension and the resulting dataset. Section IV reports the domain shift analysis and sim-to-real benchmark results. Section V concludes the paper.

II. Related Work

Public underwater datasets cover complementary but still fragmented parts of the sensing problem. UATD [7] is a forward-looking sonar benchmark based on real multibeam acquisitions, with public annotations for underwater object detection. The UXO dataset [6] instead provides real acoustic and optical sensing for underwater unexploded ordnance perception. More recently, the Marine Debris Forward-Looking Sonar Datasets [13] extended the availability of public forward-looking sonar data across multiple acquisition settings, including watertank, turntable, and flooded-quarry scenarios, and supported several tasks such as classification, detection, segmentation, and patch matching. Side-scan sonar datasets such as AI4Shipwrecks [14] and the engineering structure recognition dataset of Du et al. [15] further broaden the benchmark landscape. However, these resources remain task-specific and do not provide a simulator-side acquisition pipeline where scene population, sensing conditions, and trajectories can be regenerated programmatically while keeping RGB, sonar, and telemetry streams.

Simulation has likewise been explored through several general-purpose underwater robotics frameworks. UUV Simulator [16] established an early Gazebo-based underwater robotics environment (Gazebo is a general-purpose robot simulation platform, for underwater intervention and multi-robot simulation), explicitly modeling hydrostatic and hydrodynamic effects, thrusters, sensors, and disturbances. DAVE [17] extended this line toward a broader open-source stack for underwater robots, sensors, and environments. Stonefish was originally introduced as an open-source marine-robotics simulator with a ROS interface [18], and more recent work has shown how it can support machine-learning-oriented workflows through richer sensing and annotation capabilities [19]. At the sonar simulation level, Cerqueira et al. [20] focus on acoustic rendering realism through real-time modeling of sonar effects. That objective is complementary to ours, but different in scope. *ALAR* is not proposed as just another dataset, a general-purpose simulator, or a sonar renderer; rather, it is a runtime-configurable multimodal acquisition pipeline built on HoloOcean [9], [10], with JSON-regenerated worlds and explicit sonar refresh after scene changes so that runtime-inserted geometry remains acoustically observable.

III. The *ALAR* Extension

HoloOcean is an underwater robotics simulator built on Unreal Engine that combines vehicle dynamics, configurable sensors, and acoustic rendering within interactive 3D environments [9], [10]. In our work, it provides the simulation backbone: the execution environment in which scenes are instantiated, the ROV is moved, and multimodal observations are recorded. However, using HoloOcean as a procedural data-generation engine rather than a static scene simulator exposes two practical limitations. First, scene population requires manual intervention inside the simulator rather than external programmatic control. Second, runtime scene changes are not automatically reflected in the sonar rendering pipeline, because cached geometric structures can persist after the world has been modified. *ALAR* addresses both limitations: it adds a runtime scene-population API that drives object and clutter placement from configuration files, and an explicit sonar refresh mechanism that guarantees acoustic consistency after each scene update.

A. Runtime Scene Population

Our pipeline extends HoloOcean with runtime-configurable scene generation capabilities. The core of our extension is an API that allows object and clutter placement to be controlled externally via configuration files rather than through manual intervention. This is crucial for dataset generation, as the same acquisition logic can then be reused across different target populations, clutter densities, and layout definitions with limited operator overhead. For example, a single asset can be created using the following function (`env` is the variable encoding the simulated environment):

```
mesh_asset = "/Game/torpedo.torpedo"  # asset example

env.send_world_command(
    "SpawnAsset",
    num_params=[px, py, pz, rl, pt, yw, sx, sy, sz],
    string_params=[mesh_asset, "lb", "un"],
)
env.tick()
```

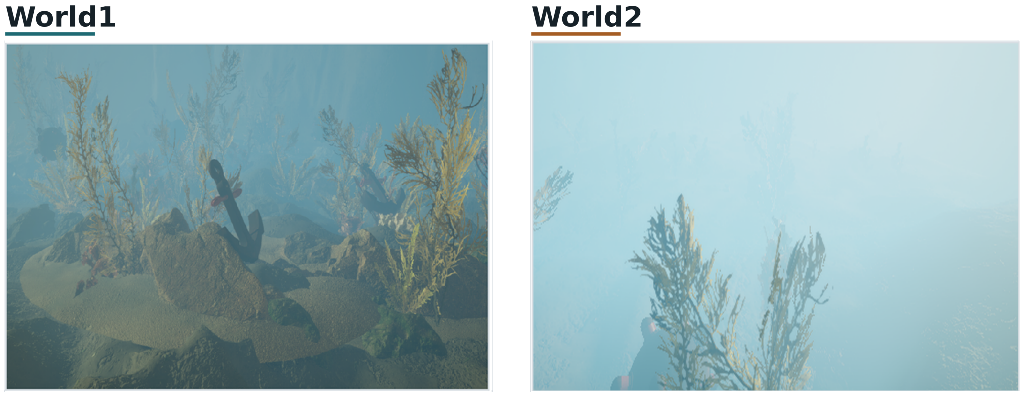

Fig. 2. Representative overview of the two world-level acquisition settings used in *ALAR*: World1, a structured object-centric world, and World2, a cluttered runtime-generated world.

Above, `mesh_asset` identifies the Unreal package path of the asset to be loaded and spawned. The parameters `px`, `py`, and `pz` denote the asset position along the three world axes; `rl`, `pt`, and `yw` denote roll, pitch, and yaw; and `sx`, `sy`, and `sz` denote per-axis scaling factors. Likewise, `lb` is the optional actor label and `un` the optional units field. Notice that, units can be interpreted either in native Unreal coordinates (`ue` or omitted) or in HoloOcean meters. This interface supports both direct insertion and scripted regeneration from JSON configurations. The experiments use custom mines, torpedoes, and anchors, developed for this work together with two seabed assets.

Runtime population is crucial whenever newly inserted acoustically relevant objects are visible to the sonar. In HoloOcean, sonar computation relies on octree-based geometric structures and cached world-state information that can persist across repeated captures within the same simulator session: if new actors are spawned after those structures have been built, the sonar uses stale geometry and therefore ignores the newly inserted assets. Our extension addresses this issue by marking the existing world geometry as stale whenever runtime actors are updated: the corresponding cached octree states are cleared and the sonar internal state is reset, forcing HoloOcean to build new octrees before acquisition resumes. This explicit invalidation and rebuild step is crucial, as it makes repeated runtime population compatible with acoustically coherent multi-run dataset generation and stable reacquisition of the same generated scene.

B. Multimodal Data Acquisition

To acquire synthetic data, we use a simulated ROV equipped with a multimodal sensor suite: a front RGB camera for forward-looking scene context, a bottom RGB camera for close-range target observation, an imaging sonar for acoustic perception under turbid conditions, and auxiliary telemetry (pose, quaternion, yaw, linear velocity, and altitude) for trajectory-aligned annotation. To probe the full range of conditions a deployed system may encounter, we organize the dataset into three complementary acquisition settings that span a controlled-to-realistic spectrum: *World1*, *World2*, and *Only Objects* (summarized in table I). The dataset associated with this paper is publicly available [12].

World1 (see fig. 2) is manually prepared, structured, and object-centric. Target objects are deliberately arranged in clear, repeatable configurations so that multiple viewpoints and orientations can be captured under controlled conditions. It combines runs from two HoloOcean environments (`DAM` and `OpenWater`) for the `seafloor`, `tube`, and `submarine` categories, and from our `Custom` environment for the `mine`, `anchor`, and `torpedo` acquisitions. Overall, World1 is balanced at the run level across these six categories (48 runs each), with four motion patterns and four scan altitudes per environment. For each nominal straight lane, two laterally shifted variants offset by ± 1 m are also included. The runs are therefore intentionally short, manually defined, and balanced: in this regime, a run is used to factorize controlled conditions and sample them uniformly across categories, motion patterns, altitudes, and lateral offsets. This design yields 288 runs and 14450 samples.

World2 (see fig. 2) is runtime-generated and more complex. Targets and distractors are inserted through code at random but deterministic positions, and the resulting layout is saved to JSON for repeatable reconstruction and reacquisition. Compared with World1, World2 introduces stronger clutter, less regular object placement, more frequent partial occlusions, and in the current scene instance some objects that are partially buried. Three long coverage runs scan 75 planned lanes at three altitudes (again with ± 1 m lateral variants), yielding 9508 samples. Here a run has a different role: each one is a long coverage mission over a cluttered runtime-generated scene, so diversity is accumulated within the run through lane sweeps rather than by multiplying short episodes; the three runs therefore match the three scan altitudes. The pronounced class imbalance visible in table I reflects the realistic dominance of seafloor returns in open-coverage sweeps, and is itself a property of interest for evaluation.

The *Only Objects* world is specifically designed to support

TABLE I
OVERVIEW OF THE CURRENT *ALAR* DATASET INSTANCE.

Folder	Runs	Samples	Acquisition profile	Key statistics
World1	288	14450	Manually arranged, structured, and object-centric acquisition with four motion patterns (straight, right lateral, left lateral, oscillatory drift) and four scan altitudes for each environment.	Sonar labels: 4800 seafloor, 4800 tube, 1104 submarine, 1227 mine, 1302 anchor, and 1217 torpedo.
World2	3	9508	Runtime-generated cluttered acquisition with 75 planned lanes, a single motion pattern (straight), and three scan altitudes; object instances are distributed with different orientations.	Sonar labels: 8450 seafloor, 482 anchor, 285 mine, and 291 torpedo frames.
Only Objects	1	3064	Object-only acquisition run with two motion patterns (straight and oscillatory drift), and five depth levels over isolated targets, without seabed/background context.	Sonar labels: 1036 anchor, 1132 mine, and 896 torpedo frames.

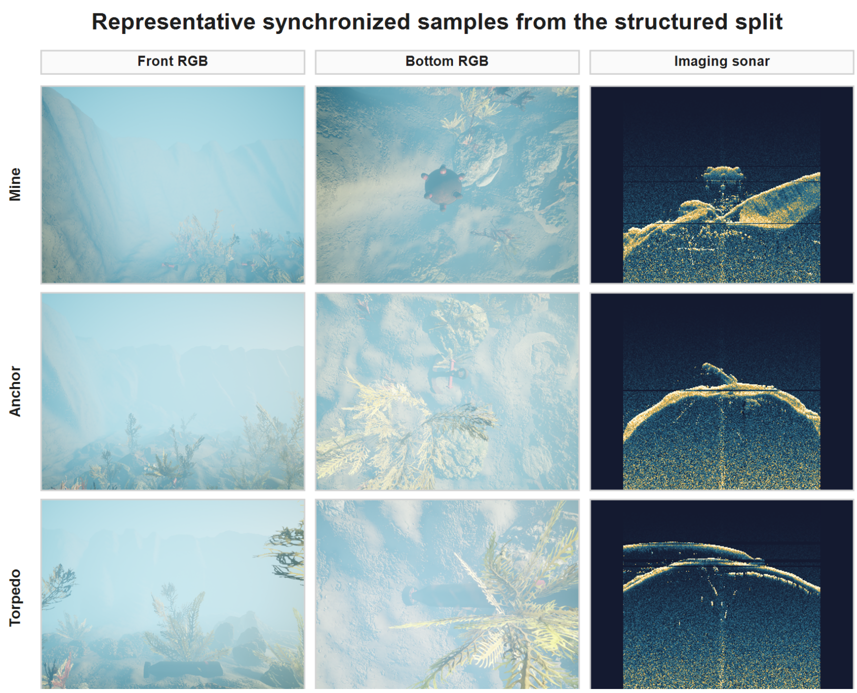

Fig. 3. Representative samples from World1. Columns show, for the same timestamp, Front RGB, Bottom RGB, and Imaging sonar; rows correspond to Mine, Anchor, and Torpedo examples.

sim-to-real transfer. To the best of our knowledge, no public dataset provides sonar observations of isolated underwater objects acquired from a moving drone platform without seabed or clutter context, making controlled synthetic-to-real alignment on target signatures alone currently intractable with real data. To fill this gap, a single scripted run executes trajectories across the three target categories (`anchor`, `mine`, `torpedo`), two lateral offsets, with $\pm 1\,\mathrm{m}$ lateral variants, and five depth levels, producing 3064 samples with perfectly aligned sonar and bottom-camera labels. The background-free scene is intentionally kept fixed to isolate target signatures for sim-to-real transfer, so adding more runs would mostly duplicate the same minimal context; diversity is instead concentrated within this single scripted run through multiple trajectories, offsets, and depth levels. By stripping away seabed returns and background clutter, this setting exposes the pure acoustic signature of each target class, providing the cleanest possible training signal for transfer to real sonar imagery.

Sample format. Each sample stores a front RGB image and a bottom RGB image as PNG files at 640×480, an imaging

TABLE II
RESNET18 RESULTS ON POLAR SONAR IMAGES.

Train	Test	Labels	BA	Macro F1	Acc.
W2	W1	4 cls	61.75	60.94	77.66
W1	W2	4 cls	40.54	36.24	75.34
W1+W2	held-out W1+W2	6 cls	97.88	97.90	97.88

sonar observation as a 256×256 float_{32} matrix in `.npz` format, and frame-level metadata in `frames.csv`. Because the bottom camera and the imaging sonar observe different spatial footprints and may only partially overlap, the metadata file records two independent annotation fields: `label` for the sonar-centred target assignment and `bottom_label` for the bottom-camera view. A derived `sonar_polar_img` entry provides a deterministic polar PNG export for image-based workflows. Run-level metadata are stored separately in `metadata.yaml`. Across all three settings, the current release contains **27022 samples over 292 runs**.

Representative samples from World1 are shown in fig. 3.

IV. BENCHMARKING *ALAR* DATA GENERATION

A. Dataset Characterisation

To verify that the generated data support supervised learning and to characterise the gap between the two acquisition regimes, we train a ResNet18 classifier on exported polar sonar images under three settings (results in table II). A mixed held-out benchmark trains on World1 and World2 jointly and tests on unseen images from both worlds (six-class label union). Two cross-world transfer benchmarks each restrict the problem to the four classes shared by both worlds (`anchor`, `mine`, `seafloor`, `torpedo`), training on one world and evaluating on the other. The mixed setting reaches 97.88% accuracy, confirming that the generated data are consistent enough to support stable learning. The cross-world results reveal an asymmetric and severe domain shift: transferring from World2 to World1 drops to 61.75% balanced accuracy, while the reverse direction collapses to 40.54% balanced accuracy and 36% macro F1. The apparent 75.34% accuracy in this direction is misleading: the model defaults to predicting the seafloor majority class and is only exposed by the class-aware metrics. The asymmetry is informative: a model trained on clean, object-centric World1 data has no exposure to clutter or coverage-sweep geometry, and therefore fails almost completely on the realistic World2 distribution. Together, the three results confirm that World1 and World2 are structurally distinct and that clutter and acquisition geometry introduce a unidirectional difficulty gradient that cannot be resolved by treating the two worlds as a single homogeneous corpus.

B. Zero-Shot Transfer to Real Sonar Imagery

To evaluate zero-shot transfer, we test a ResNet18 trained solely on synthetic *ALAR* data against real polar sonar images from the public UXO dataset [6], with no real-data training at any stage. That dataset focuses on bomb categories, so we evaluate transfer on the two elongated classes that are visually and acoustically closest to our simulated torpedo asset: `15cm_mortar` and `100lbs_bomb`. Training uses only the synthetic Only Objects folder, which contains `anchor`, `mine`, and `torpedo` samples without seabed context. On the synthetic validation split from this folder, ResNet18 reaches 97.21% accuracy and 97.22% macro F1, confirming that the network learns stable class signatures before transfer. We then evaluate directly on 28663 real UXO images (13899 `15cm_mortar` and 14764 `100lbs_bomb`), with results reported in table III.

TABLE III
RESNET18 SIM-TO-REAL RESULTS, SEPARATED INTO SYNTHETIC VALIDATION AND REAL UNSEEN TEST DATA.

Phase / Category	Images	Metric	Value (%)
Validation (synthetic held-out)			
Synthetic held-out	538	Accuracy	97.21
Synthetic held-out	538	Macro F1	97.22
Test (real unseen UXO images)			
15cm mortar	13899	Recall (torpedo)	90.75
100lbs bomb	14764	Recall (torpedo)	56.92
Real overall	28663	Recall (torpedo)	73.33

Without any exposure to real sonar imagery during training, the model reaches 73.33% overall torpedo recall, with substantially stronger transfer to `15cm_mortar` (90.75%) than to `100lbs_bomb` (56.92%). This asymmetry suggests that the simulated torpedo asset captures the acoustic signature of more slender elongated targets more faithfully than broader ones.

V. CONCLUSION AND FUTURE WORK

We presented *ALAR*, a runtime-configurable pipeline for multimodal underwater dataset generation, built on two technical enablers: runtime scene adaptation through custom asset insertion, and an explicit sonar refresh mechanism that keeps runtime-inserted geometry acoustically observable after scene changes. Together these enable what static simulators cannot: the generation of new scene configurations without manual simulator editing, followed by reacquisition with stable and acoustically coherent sonar observations across arbitrary target configurations. The resulting dataset spans a controlled-to-realistic spectrum across three acquisition settings, and the ResNet18 experiments confirm that the generated data are learnable, that World1 and World2 are structurally distinct rather than interchangeable, and that synthetic torpedo signatures transfer non-trivially to real UXO sonar imagery without any real-data training.

The greatest threat to validity of the current work is that sim-to-real validation remains partial: only torpedo-like targets could be matched to an existing real dataset, and the real images were not acquired with a moving drone platform. The critical next step is therefore a paired real-world acquisition campaign with a physical ROV carrying the same sensor stack and scanning physical counterparts of the same mine, anchor, and torpedo assets used in simulation. This alignment of target geometry, category structure, and annotation logic across synthetic and real acquisition would turn *ALAR* from a promising sim-to-real starting point into a closed-loop benchmark.

References

[1] M. Grimaldi, D. Nakath, M. She, and K. Köser, "Investigation of the challenges of underwater-visual-monocular-slam," *ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, vol. X-1/W1-2023, pp. 1113–1121, Dec. 2023. [Online]. Available: http://dx.doi.org/10.5194/isprs-annals-X-1-W1-2023-1113-2023

[2] F. F. R. Merveille, B. Jia, Z. Xu, and B. Fred, "Advancements in sensor fusion for underwater SLAM: A review on enhanced navigation and environmental perception," *Sensors*, vol. 24, p. 7490, 2024.

[3] B. Kim, H. Joe, and S.-C. Yu, "High-precision underwater 3D mapping using imaging sonar for navigation of autonomous underwater vehicle," *Int. J. Control Autom. Syst.*, vol. 19, no. 9, pp. 3199–3208, 2021. [Online]. Available: https://doi.org/10.1007/s12555-020-0581-8

[4] Z. Wang, J. Guo, S. Zhang, and Y. Zhang, "Sonar-based object detection for autonomous underwater vehicles in marine environments," *Frontiers in Marine Science*, vol. 12, p. 1539371, Apr. 2025.

[5] M. Jian, N. Yang, C. Tao, H. Zhi, and H. Luo, "Underwater object detection and datasets: a survey," *Intelligent Marine Technology and Systems*, vol. 2, p. 9, 2024.

[6] N. Dahn, M. B. Firvida, P. Sharma, L. Christensen, O. Geisler, J. Mohrmann, T. Frey, P. K. Sanghamreddy, and F. Kirchner, "An acoustic and optical dataset for the perception of underwater unexploded ordnance (UXO)," in *OCEANS 2024 - Halifax*, 2024.

[7] K. Xie, J. Yang, and K. Qiu, "A dataset with multibeam forward-looking sonar for underwater object detection," *Scientific Data*, vol. 9, no. 1, 2022. [Online]. Available: http://dx.doi.org/10.1038/s41597-022-01854-w

[8] N. Jaber, Q. Al-Ramahi, L. Christensen, and B. Wehbe, "Sonarcloud: A simulated forward looking sonar dataset for underwater perception tasks," in *OCEANS 2025 MTS/IEEE Conference*, Chicago, IL, USA, 2025.

[9] E. Potokar, S. Ashford, M. Kaess, and J. Mangelson, "HoloOcean: An underwater robotics simulator," in *Proc. IEEE Intl. Conf. on Robotics and Automation (ICRA)*, Philadelphia, PA, USA, May 2022.

[10] E. Potokar, K. Lay, K. Norman, D. Benham, T. Neilsen, M. Kaess, and J. Mangelson, "HoloOcean: Realistic sonar simulation," in *Proc. IEEE/RSJ Intl. Conf. on Intelligent Robots and Systems (IROS)*, Kyoto, Japan, Oct. 2022.

[11] A. Bedei, "Andreabedei1/holooceanspawner: Holooceanspawner," Mar. 2026. [Online]. Available: https://doi.org/10.5281/zenodo.19231430

[12] A. Bedei, L. Bacchiani, A. Melis, R. Girau, F. Callegati, and G. Pau, "ALAR: A multimodal underwater dataset for harsh-domain perception," 2026. [Online]. Available: https://dx.doi.org/10.21227/qd5q-rp93

[13] M. Valdenegro-Toro, D. C. Padmanabhan, D. Singh, B. Wehbe, and Y. Petillot, "The marine debris forward-looking sonar datasets," in *OCEANS 2025 Brest*, 2025, pp. 1–10.

[14] A. V. Sethuraman, A. Sheppard, O. Bagoren, C. Pinnow, J. Anderson, T. C. Havens, and K. A. Skinner, "Machine learning for shipwreck segmentation from side scan sonar imagery: Dataset and benchmark," *The International Journal of Robotics Research*, vol. 44, no. 3, pp. 341–354, 2025.

[15] X. Du, Y. Sun, Y. Song, L. Dong, C. Tao, and D. Wang, "Recognition of underwater engineering structures using CNN models and data expansion on side-scan sonar images," *Journal of Marine Science and Engineering*, vol. 13, no. 3, p. 424, 2025.

[16] M. M. M. Manhães, S. A. Scherer, M. Voss, L. R. Douat, and T. Rauschenbach, "Uuv simulator: A gazebo-based package for underwater intervention and multi-robot simulation," in *OCEANS 2016 MTS/IEEE Monterey*, 2016, pp. 1–8.

[17] M. M. Zhang, W.-S. Choi, J. Herman, D. Davis, C. Vogt, M. McCarrin, Y. Vijay, D. Dutia, W. Lew, S. Peters, and B. Bingham, "Dave aquatic virtual environment: Toward a general underwater robotics simulator," 2022. [Online]. Available: https://arxiv.org/abs/2209.02862

[18] P. Cieślak, "Stonefish: An advanced open-source simulation tool designed for marine robotics, with a ROS interface," in *OCEANS 2019 - Marseille*, 2019, pp. 1–6.

[19] M. Grimaldi, P. Cieślak, E. Ochoa, V. Bharti, H. Rajani, I. Carlucho, M. Koskinopoulou, Y. R. Petillot, and N. Gracias, "Stonefish: Supporting machine learning research in marine robotics," in *2025 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2025, pp. 13 605–13 611.

[20] R. Cerqueira, T. Trocoli, J. Albiez, and L. Oliveira, "A rasterized ray-tracer pipeline for real-time, multi-device sonar simulation," *Graphical Models*, vol. 111, p. 101086, 2020. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1524070320300278