Robotic Detection of a Human-Comprehensible Gestural Language for Underwater Multi-Human-Robot Collaboration

Sadman Sakib Enan¹, Michael Fulton² and Junaed Sattar³

Abstract—In this paper, we present a motion-based robotic communication framework that enables non-verbal communication among autonomous underwater vehicles (AUVs) and human divers. We design a gestural language for AUV-to-AUV communication which can be easily understood by divers observing the conversation — unlike typical radio frequency, light, or audio-based AUV communication. To allow AUVs to visually understand a gesture from another AUV, we propose a deep network (RRCommNet) which exploits a self-attention mechanism to learn to recognize each message by extracting maximally discriminative spatio-temporal features. We train this network on diverse simulated and real-world data. Our experimental evaluations, both in simulation and in closed-water robot trials, demonstrate that the proposed RRCommNet architecture is able to decipher gesture-based messages with an average accuracy of 88-94% on simulated data and 73-83% on real data (depending on the version of the model used). Further, by performing a message transcription study with human participants, we also show that the proposed language can be understood by humans with an overall transcription accuracy of 88%. Finally, we discuss the inference runtime of RRCommNet on embedded GPU hardware, for real-time use on board AUVs in the field.

I. INTRODUCTION

Over the last several decades, applications of autonomous underwater vehicles (AUVs) [1], [2] have multiplied and diversified (e.g., environmental monitoring and mapping [3], [4], submarine cables and wreckage inspection [5], search and navigation [6], [7]), driven by ever-increasing on-board computational power, increased affordability, and ease of use. The majority of these applications involve multiple AUVs and/or their human diver companions, often interacting with one another to work effectively as a team [8], [9]. Thus, robust underwater human-to-robot and robot-to-human interaction capabilities are of utmost value. A common language comprehensible to both humans and other AUVs would greatly enhance such underwater multi-human-robot (m/HRI) missions (see Fig. 1).

When designing such a communication protocol, challenges unique to the underwater domain need to be considered. Traditional sensory mediums, such as radio and other electromagnetic (EM) modalities, suffer from signal attenuation and degradation [10] underwater which limit their use to mostly surface operations. Although acoustic signals work quite well in underwater settings [11], these types of inter-AUV communication signals are typically incomprehensible to humans. Our recent work utilizing robot motion for AUV-to-diver communication has demonstrated that motion can be used to communicate with divers [12], [13]. Similarly, research on the use of motion for inter-AUV communication has shown the same for AUV-to-AUV communication [14]. However, the two capabilities have yet to be combined in a single language for multi-human-robot communication. Visual perception of gestures is the natural choice for AUVs, as many of them are equipped with low-cost vision sensors. While vision underwater can be impacted by water quality and turbidity, promising results achieved in the improvement of underwater vision (e.g., [15], [16]) provide increased robustness in AUV visual perception.

To create a natural, comprehensible, and accurate language for robots in an m/HRI context, we propose to use robot motion to design gestural messages as in [12], [13], for both robot-to-human and robot-to-robot communications (see Fig. 2). We design these messages in simulated underwater environments [17], using computer-aided design (CAD) renderings of a six-legged AUV named Aqua [18]. Additionally, we implement the gestural messages on board an actual Aqua robot using the Robot Operating System (ROS) [19]. For a robot to interpret these messages, we propose a recognition network, Robot-to-Robot Communication Network (RRCommNet), which learns salient spatio-temporal features from the gestural messages using a self-attention mechanism. After training RRCommNet on simulated and real-world data, our experiments show the recognition accuracy of RRCommNet to be approximately 94% on simulated data and 83% on real data (closed water pool environment). Based on our previous experience with computer vision-based methods in underwater field environments (e.g., [15], ...
[20]), we are aware of the challenges in transitioning from simulated and controlled to field environments. However, training and evaluation in these environments is a necessary prerequisite to the creation and evaluation of a system in the field. We also show that we can improve the inference time of RRCommNet at a small cost to accuracy (simulated: 88%, real: 73%) by down-sampling the input video by half. Finally, through a transcription experiment, we show that humans can comprehend a conversation between two AUVs using the gestural messages with a transcription accuracy of 88%. Together, these two evaluations demonstrate that our gestural communication system can be used for accurate communication by an AUV in an m/HRI context. Thus, in this paper, we:

1) Propose a gestural language for AUV-to-AUV communication,
2) Create an end-to-end gestural message recognition network, RRCommNet, to interpret underwater communication between AUVs,
3) Perform experiments in both simulated and real underwater environments to validate the performance of the proposed gestural message recognition network, and
4) Conduct a study which demonstrates that the proposed gestural language can be understood by humans.

II. RELATED WORK

Underwater communication has largely focused on human-to-robot communication, i.e., regulating underwater robots based on human inputs. One common approach is to control the robot via high-speed tethered communication [21], however, direct communications are often preferred in missions as opposed to using tethered connections. A number of direct communication techniques do not even require additional robot hardware, e.g., using fiducial markers [22] or hand gestures [20], [23]. For robot-to-human communication, on the other hand, the use of small displays is by far the most popular method [24] although it has significant limitations in readability at distance, at an angle, and under low water quality. There are alternatives, such as the use of a bi-directional communication device [25] and the use of light to represent simple ideas [26], however, they either require the addition of dedicated communication devices or lack functionalities. In comparison, the use of robot motion for information [12] or affective displays for appearance-constrained robots [27] has seen encouraging results in communicating with humans.

In contrast to human-in-the-loop communication, underwater robot-to-robot communication systems are significantly less studied, primarily due to the fact that robots are not as perceptive as humans and therefore need sophisticated algorithms to understand what other robots are trying to communicate. In [28], body markings (helical drawings) are used by AUVs to communicate relative pose information. More recently, Koreitem et al. [14] propose a communication system for underwater robots where the communication messages are represented using full-body gestures and optimal variable-length prefix codes. However, this technique involves multiple steps to devise different messages and the proposed CNN learns indirectly from them. As a result, adding additional messages is not straightforward. In this work, we focus on methods such as activity recognition that can directly learn from gesture-based messages.

Activity recognition (AR) is a well-studied problem in computer vision and robotics, with research spanning over two decades [29], [30], [31]. The goal is to predict an activity class from a large pool of human activities involving exercise, sport, instrument performance, everyday life, etc. The general solution to this problem is to learn robust spatio-temporal features from different activity classes which are contained in small video clips. A variety of methods have been used to integrate the temporal information with the spatial, such as pooling [32], fusion [33], recurrent [34], two-stream [35], [36], and 3D architectures [37]. These techniques perform well mainly because of their ability to learn from publicly available large datasets involving human activities. The lack of such datasets is the primary contributing factor behind the absence of activity recognition techniques for robot actions in the literature. Therefore, robotic researchers often use synthetic data to validate their proposed methods before fine-tuning them for real robotic platforms [38], [39]. Furthermore, unlike traditional human activity recognition datasets, the communication messages we have designed for underwater robots have high similarities in terms of background activities and the overall appearance of the robot motions. Additionally, there are cases in which a portion of a gesture’s motion can be common to multiple gestures, leading to a strong resemblance between two gestures when considering small portions of their motion. As a result, AR algorithms that perform well for human activity recognition may not work well for underwater robots. Recent research on natural language processing suggests that one can effectively learn the underlying sequential relations by using a self-attention mechanism [40], which has also shown encouraging results in AR recently [41], [42]. Inspired by such work, we believe that highly robust spatio-temporal features can
be learned from our gesture-based communication messages using a similar attention mechanism.

III. GESTURAL LANGUAGE RECOGNITION FRAMEWORK

A. Communication Message Design

Since the purpose of using motion-based gestural communication is to enable human understanding of robot-to-robot conversations, we draw inspiration from our previous work on robot-to-human underwater communication methods [12], [13], where we have shown that humans are able to identify gestural messages with reasonable accuracy. We create a library of communication messages based on our experience working with AUVs in real underwater environments and discussions with potential end-users (e.g., marine biologists). Our library includes three types of messages: 1) Directional (D): which relate to a notification or command of some directional movement, 2) Information/Command (I/C): which provide information or give direct commands to another robot, and 3) Conversation Control (CC): which are responses to questions or commands. Table I shows the complete library of communication messages with their description, type, definition, and average duration.

B. Gesture Implementation

We first implement the gestural messages in simulated underwater environments using Unreal Engine visual programming [17], with CAD renderings of the Aqua robot. We use three self-made environments: pool, lake, ocean, and a separate ready-made ocean environment [43] as the visual environments the gestures will be performed in. To implement the messages in real underwater environments with the Aqua robot, we create a ROS package named ecomm which operates as a gestural message generator. The generator can produce 15 communication messages using a user-defined configuration file, an example of which can be seen in Fig. 3 for the START MAPPING message. The configuration file contains a definition of the gesture as a set of timed motions expressed in roll, pitch, yaw, surge, and heave velocity percentages. Our ROS node parses these configuration files to implement the robot maneuvers using the robot motion controller [44].

C. Gestural Message Dataset

To train a neural network capable of recognizing our gestural messages, we first create a dataset by recording the 15 defined gestural messages (see Table I) as full-color videos in both simulated and real underwater environments. To improve the robustness of our simulated training data, we introduce a number of variations within the simulated environment and the robot model, giving rise to 25 different environmental conditions. Specifically, we vary the surface texture, object type, hydrodynamics, water visibility, and color of the robot while keeping the message definitions unchanged. For real-world data, we record 5 instances of each message performed in a closed-water underwater environment. For the HELP message, the Aqua robot’s PID controller was overshooting the requested target angles, causing the robot to stray away from the defined circular path.

---

**Table I**: Listing of communication messages [D=Directional, I/C=Information/Command, CC=Conversation Control].

| Communication Message | Description and Type | Definition | Average Duration (secs) |
|-----------------------|----------------------|------------|------------------------|
| BATTERY LOW           | Signal low battery (I/C) | Roll 360° twice while moving down vertically | 4.28 |
| START COMMUNICATION   | Begin robot-to-robot communication (CC) | Roll, pitch, and yaw 45° at the same time twice | 16.71 |
| ASCEND                | Go up to a certain depth (D) | Pitch up 90° vertically and go up | 3.18 |
| DESCEND               | Go down to a certain depth (D) | Pitch down 90° vertically and go down | 3.28 |
| FOLLOW ME             | Instruct another robot to follow it (D) | Pitch and yaw 35° to the right, comeback to original position; do this twice and then roll 45°, make a 180° turn while unrolling and finally, go forward | 6.93 |
| DANGER                | Danger nearby (I/C) | Yaw left and right thrice; then roll 45°, make a 180° turn while unrolling | 7.44 |
| COLLECT DATA          | Start data collection (I/C) | Pitch and yaw 45° to the right, comeback to original position, do the same to the left; perform this twice | 8.32 |
| START MAPPING         | Map the environment (I/C) | Yaw 360° while pitching up and down | 4.32 |
| GO TO LOCATION        | Go to a specific location (D) | Roll 45°, move forward and backward twice, then go to a location | 7.48 |
| U-TURN                | Danger nearby, make a u-turn (D) | Roll 80° and make a 180° turn while unrolling | 4.22 |
| HELP                  | Call for help (I/C) | Roll 45°, circle in a small loop twice | 19.90 |
| EMERGENCY SURFACING   | Go to the surface of the water (D) | Pitch up 90° and roll 360° twice | 3.46 |
| STOP                  | Stop doing whatever you’re doing (CC) | Yaw 360° twice before stopping | 6.52 |
| NO                    | Disagreement (CC) | Yaw left and right twice | 4.31 |
| YES                   | Agreement (CC) | Pitch down and up twice | 4.26 |
We use PyTorch [47] libraries to implement RRCommNet, with input spatial resolution of 320 × 256 and temporal information. The second and more significant issue is the commonality between portions of pairwise gestures; a strong resemblance can confuse the networks when considering small durations of such pairs of gestures. We address this by using a self-attention mechanism (proposed in [40]) with bi-directional Transformers (BERT [46]), as was suggested in [41], [42]. With self-attention, the network will be able to focus on the most salient features while BERT enables temporal information from both directions. Taking these together, the network is able to look into the most salient features within a longer temporal limit enhancing the messages’ contexts. With these choices in mind, we design RRCommNet (see Fig. 4) as described below.

First, we take a query feature \( Q \) from the extracted feature representation (computed using ResNeXt-101) to compare against all other positions in the feature representation (considered as key, \( K \)). For a gestural message, \( Q \) is the small spatial information with a large temporal context. Both query and key are of size \( d_{model}(= C') \). The comparison gives rise to a weighted sum of values \( V \) which are the self-attention scores. Note that to learn the robust feature representations of \( Q \) and \( K \), we use linear projection to project them to a lower dimension \( d_k \). Specifically, we compute the self-attention as follows:

\[
A_{self} = \frac{QK^T}{\sqrt{d_k}} \\
A_{weights}^i = \frac{\exp(a_{self}(:, i))}{\sum_{j=1}^{d_k} \exp(a_{self}(:, j))} \\
A_{self}^i = A_{weights}^iV
\]

where, \( A_{self} \) is the weighted self-attention scores.

Note that we use the attention mechanism in a parallel fashion, i.e., using multiple attention heads (\( h \)), and set \( d_k = \text{floor}(d_{model}/h) \). The multi-head attention scores are fed through a two layer position-wise feed-forward network (P-FFN) to get the final self-attention scores to represent the salient features from a gestural message. P-FFN is defined as, \( \text{P-FFN}(A_{self}) = \max(0, A_{self}W_1 + b_1)W_2 + b_2 \), where \( W_i \) and \( b_i \) are the weights and biases of the respective layers.

Finally, a single linear layer is used to get the final classification scores which has exactly \( N \) neurons, where \( N \) is the total number of communication messages. Note that we perform the final prediction using the classification embedding which we include on top of the feature representation for better classification, as suggested in [46], [42]. Additionally, we include a location embedding to all the locations in the feature representation in order to incorporate positional information in the attention scores. These embeddings are set as learnable parameters. Moreover, while performing the self-attention mechanism, we randomly mask 10% of the feature locations and set their attention scores to zeros which incorporates the bi-directional context learning from BERT.

**E. Implementation Details**

We use PyTorch [47] libraries to implement RRCommNet, with input spatial resolution of 320 × 256 and temporal
TABLE II: Results of the human transcription study in recognizing different gestural messages. The values represent avg. trans. accuracy (%) and confidence (out of 10).

| Gest. Message | BATTERY | START | COMM. | ASCEND | DESCEND | FOLLOW | ME | DANGER | COLLECT | DATA | START | MAPPING |
|---------------|---------|-------|-------|--------|---------|--------|----|--------|---------|------|-------|---------|
| Human Trans. | 94.10/8.30 | 94.10/7.30 | 97.10/8.30 | 94.10/8.30 | 97.10/8.60 | 72.10/7.80 | 85.30/8.30 | 58.80/7.60 |

Table II shows the results of the study. As we can see from the table, the participants are able to correctly transcribe the conversations shown to them in a simulated video of two robots conversing. A total of 10 such conversations are shown from one of three different viewpoints: head-on, rotated by 90 degrees of yaw, and rotated by 90 degrees of pitch. Our population of 34 participants are randomly assigned to one of these viewpoints, asked to select the displayed gesture’s meaning from a drop-down list for each message, and rate their confidence in their answers for each transcription from zero to 10. Table II shows the results of the study. As we can see from the table, the participants are able to correctly transcribe the conversation with an average accuracy of 88.20% and confidence of 7.9 (out of 10). Here, avg. accuracy = \( \frac{1}{34} \sum_{p=1}^{34} \left( \frac{\text{correct selections}}{\text{total shown}} \right) \). The participants seem to struggle the most with the messages which RRCommNet performs poorly on (discussed in Sec. V-B). Closer inspection of such messages indicates that they include gestures which have high visual similarity. The results from this study show that humans can comprehend our proposed gestural language, which, along with the results in the following section, fulfills our goals of a gestural language that can be understood by both robots and humans.

IV. HUMAN TRANSCRIPTION STUDY

To demonstrate that the proposed gestural language can be effectively understood by humans in an m/HRI mission, we undertake a study\(^1\) of humans transcribing the conversations of robots using our gestural language, using the QualtricsXM survey platform. First, the participants are taught the messages in a random order. Then, they are asked to transcribe the conversation shown to them in a simulated video of two robots conversing. A total of 10 such conversations are shown from one of three different viewpoints: head-on, rotated by 90 degrees of yaw, and rotated by 90 degrees of pitch. Our population of 34 participants are randomly assigned to one of these viewpoints, asked to select the displayed gesture’s meaning from a drop-down list for each message, and rate their confidence in their answers for each transcription from zero to 10. Table II shows the results of the study. As we can see from the table, the participants are able to correctly transcribe the conversation with an average accuracy of 88.20% and confidence of 7.9 (out of 10). Here, avg. accuracy = \( \frac{1}{34} \sum_{p=1}^{34} \left( \frac{\text{correct selections}}{\text{total shown}} \right) \). The participants seem to struggle the most with the messages which RRCommNet performs poorly on (discussed in Sec. V-B). Closer inspection of such messages indicates that they include gestures which have high visual similarity. The results from this study show that humans can comprehend our proposed gestural language, which, along with the results in the following section, fulfills our goals of a gestural language that can be understood by both robots and humans.

V. EVALUATION OF RRCOMMNET

A. Evaluation Procedure and Metrics

The test videos are processed as chunks of either 64 or 32 frames (with skipping). We make the final prediction\(^2\) on 10 different cropped versions of the input video clip. We follow the same cropping mechanism as described in [48], i.e., four crops from the corners, one from the center, and the same on a flipped version. The crops have a spatial dimension of 112 × 112. Therefore, a single RGB test batch is a 5-dimensional tensor of shape [10, 64, 3, 112, 112] or [10, 32, 3, 112, 112] (with skipping).

The test batch is fed to the trained RRCommNet model to get 10 prediction scores as a tensor (\(x_{\text{preds}}\)) of shape (10, 15). We average the scores to get our final prediction vector, \(x_{\text{mean}}\). Finally, we calculate the predicted class probability using the softmax function, as shown below:

\[
P(x_{\text{mean}}^i) = \frac{\exp(x_{\text{mean}}^i)}{\sum_{j=1}^{N} \exp(x_{\text{mean}}^j)}
\]

where \(i\) refers to the \(i\)-th class, \(N\) is the total number of communication messages, and \(i = 1, 2, \ldots, N\). Each

\(^1\)Study (reference no. 00012959) has been reviewed and approved by the University of Minnesota’s Institutional Review Board.

\(^2\)In this paper, we use prediction and recognition interchangeably.
prediction class is found by selecting the index of the maximum class probability.

We consider three metrics for quantitative evaluation of RRCommNet:

1) Recognition Accuracy: it is the ratio between correct predictions and total instances (for each message).

2) Recognition Probability: it is the softmax probability, as defined in (1), that shows the confidence for a correct prediction (for each message).

3) Inference Time: it is defined as the time it takes for each prediction (reported on CPU unless otherwise specified).

B. Results

1) RRCommNet Outperforms SOTA For AUV Gestures: First, we compare the performance of the RRCommNets against the state-of-the-art (SOTA) action recognition models in terms of average recognition accuracy. From Table III, we see that for simulated data, RRCommNet achieves an average recognition accuracy of 94.67% which is exactly same as the SOTA model, LateTemporal3D Bert [42]. In comparison, the SlowFast [36] model, which is another robust action recognition model, does not achieve comparable performance (accuracy 74.67%) against our method. As for real data, RRCommNet achieves an average recognition accuracy of 83.33% which is significantly better than the rest of the methods. In contrast, RRCommNet-Skip achieves an average recognition accuracy of 88% on simulated data which is higher than the SlowFast method but only comparable against the SOTA or RRCommNet. For real data, however, RRCommNet-Skip shows superior performance than the SOTA. As a note, we were unable to make a comparison with the underwater inter-robot communication framework presented in [14] because the authors in that paper use a 3D pose regressor as the visual decoder whereas we use a self-attention based classifier as our visual decoder.

2) RRCommNet Confusion Matches Human Confusion: First, we evaluate the message recognition performance of both RRCommNet and RRCommNet-Skip by feeding our test data (described in Section III-C) to both networks and analyzing the outputs. Fig. 5 shows a snapshot of RRCommNet predicting the U-TURN message, performed by the Aqua AUV in actual underwater environment. The predictions are accumulated in a confusion matrix and further augmented by the human transcription of robotic conversation results (described in Sec. IV). Fig. 6 shows the complete confusion matrix where the values in different cells are normalized (refer to the color bar to see the non-normalized values). From the figure, we see that DANGER and START MAPPING have a resemblance to NO and STOP, respectively. Noting this, we suggest that pre-deployment analysis of spatial similarity of gestures could optimize both robot and human comprehension of gestures by avoiding overlaps in spatio-temporal fragments.

3) RRCommNet-Skip Outperforms RRCommNet in Speed: Finally, we evaluate the performance of the RRCommNets in terms of their prediction confidence and speed. From Table IV, we see that both RRCommNets are fairly confident while making gestural message predictions, having overall average recognition probability of 78.97% and 78.88%, respectively. As for recognition speed, we see that RRCommNet-Skip is notably faster than RRCommNet. As a matter of fact, both the networks display fast inference times on CPU. For example, inference times for BATTERY LOW and STOP messages are 0.55s and 1.32s, respectively using RRCommNet and 0.34s and 0.90s, respectively using RRCommNet-Skip. Therefore, we can choose RRCommNet-Skip where inference time is important, and choose RRCommNet where accuracy is the main requirement.

C. Runtime Performance on Embedded GPU

As established in the previous section, there is a trade-off between inference time and accuracy with RRCommNet and RRCommNet-Skip, as shown in Table IV. However, these inference times are on a powerful CPU (an Intel® Xeon® E5-2650), processing power that most AUVs will
TABLE IV: Comparison between the performance of RRCommNet and RRCommNet-Skip in recognizing different gestural messages. The values represent avg. recog. probability (%) / avg. inference time (s).  

| Gest. Message | BATTERY | START | COLLECT | FOLLOW | DANGER | LOCATION | STOP | NO | YES | Overall |
|---------------|---------|-------|---------|--------|--------|----------|------|----|-----|---------|
| RRCommNet     | 95.53/0.55 | 78.16/3.48 | 68.49/3.07 | 69.6/0.35 | 63.65/1.61 | 66.9/1.32 | 60.18/0.90 | 78.97/1.45 |
| RRCommNet-Skip| 95.78/0.34 | 80.93/2.37 | 70.74/3.23 | 76.29/0.35 | 37.68/0.94 | 73.1/0.67 | 69.43/0.66 | 78.88/0.80 |

not have available on-board. It is, therefore, worthwhile to consider runtime performance on a processor more likely to be used in an AUV, such as the Nvidia TX2 (already in use in a number of AUVs, Aqua among them). To test on the TX2, we simply run our evaluation experiments a second time, recording the inference time for each video, calculating the per-frame inference time for both networks, and averaging across all videos. RRCommNet operates at an average inference speed of **10 FPS** and RRCommNet-Skip performs inference at **12.5 FPS**. Total inference time for any gesture depends on the length of the gesture, but the rate of inference allows the recognition of gestures at sufficient speeds to be useful in the field.

VI. CONCLUSION

In this work, we have presented the design and implementation of a unified gestural language for underwater m/HRI missions and shown that an attention mechanism-based deep network can recognize a gestural language for AUVs in both simulated and real environments. Our networks, RRCommNet and RRCommNet-Skip, demonstrated recognition accuracy which outperforms SOTA activity recognition methods. Both networks also have inference speeds on embedded GPU which are sufficient for on board robot deployments, though RRCommNet-Skip is somewhat faster. Through a human transcription study, we have also demonstrated that the proposed gestural language is understandable to humans. Having shown our framework to be comprehensible to both humans and robots, we also suggest that our framework is ideal for integration into existing diver interaction workflows. The majority of underwater communication between divers happens via gestures (with some exceptions). In much the same way, human-to-robot control methods underwater often utilize gestures [20], [23]. Thus, a gesture-based robot-to-robot language fits nicely into the overall workflow of underwater communication. On the technical side, our ROS-based implementation makes the process of integrating our method into any ROS-powered AUV relatively straightforward. Lastly, implementation of the gestures in our language via simple configuration files makes the transition from designing a gesture to implementing it as simple as writing out the desired movements. While programming experience will still be required to fine-tune the movements, this implementation will allow for the easy creation of new gestures when they are required. Taken together, our method’s SOTA recognition performance, reasonable inference times, human comprehensibility, conformity with existing robot and human underwater communication methods, and easy generation/reconfiguration of gestures make it an excellent candidate for deployment in the field. This enables the opportunity to use our gestural language and language recognition network for AUV-to-AUV communication in underwater m/HRI missions, so that every party to the interaction, whether human or robot, can understand every part of the conversation.

REFERENCES

[1] J. Sattar, G. Dudek, O. Chiu, I. Rekleitis, P. Giguere, A. Mills, N. Plamondon, C. Prahas, Y. Girdhar, M. Nahon et al., “Enabling Autonomous Capabilities in Underwater Robotics,” in 2008 IEEE/RSJ International Conference on Intelligent Robots and Systems. IEEE, 2008, pp. 3628–3634.

[2] C. Edge, S. S. Enan, M. Fulton, J. Hong, J. Mo, K. Barthelemy, H. Bashaw, B. Kallevig, C. Knutson, K. Orpen, and J. Sattar, “Design and Experiments with LoCO AUV: A Low Cost Open-Source Autonomous Underwater Vehicle*,” in 2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), 2020, pp. 1761–1768.

[3] M. Fulton, J. Hong, M. J. Islam, and J. Sattar, “Robotic Detection of Marine Litter Using Deep Visual Detection Models,” in 2019 International Conference on Robotics and Automation (ICRA). IEEE, 2019, pp. 5752–5758.

[4] N. Weidner, S. Rahman, A. Q. Li, and I. Rekleitis, “Underwater Cave Mapping using Stereo Vision,” in 2017 IEEE International Conference on Robotics and Automation (ICRA). IEEE, 2017, pp. 5709–5715.

[5] B. Bingham, B. Foley, H. Singh, R. Canilli, K. Delaporta, R. Eustice et al., “Robotic Tools for Deep Water Archaeology: Surveying an Ancient Shipwreck with an Autonomous Underwater Vehicle*,” Journal of Field Robotics (JFR), vol. 27, no. 6, pp. 702–717, 2010.

[6] K. Koreitem, F. Shkurti, T. Manderson, W.-D. Chang, J. C. Gamboa Higuera, and G. Dudek, “One-Shot Informed Robotic Visual Search in the Wild,” in 2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), 2020, pp. 5800–5807.

[7] M. Xanthidis, N. Karapetyan, H. Damron, S. Rahman, J. Johnson, A. O’Connell, J. M. O’Kane, and I. Rekleitis, “Navigation in the Presence of Obstacles for an Agile Autonomous Underwater Vehicle*,” in 2020 IEEE/RSJ International Conference on Robotics and Automation (ICRA). IEEE, 2020, pp. 892–899.

[8] M. J. Islam, M. Ho, and J. Sattar, “Understanding Human Motion and Gestures for Underwater Human-Robot Collaboration,” Journal of Field Robotics (JFR), pp. 1–23, 2018.

[9] J. Hong, S. S. Enan, C. Morse, and J. Sattar, “Visual Diver Face Recognition for Underwater Human-Robot Interaction,” arXiv preprint arxiv:2011.09556, 2020.

[10] U. M. Qureshi, F. K. Shaikh, Z. Aziz, S. M. Z. Shah, A. A. Sheikh, E. Felenban, and S. B. Qaisar, “RF Path and Absorption Loss Estimation for Underwater Wireless Sensor Networks in Different Water Environments,” Sensors, vol. 16, no. 6, p. 890, 2016.

[11] N. Farr, A. Bowen, J. Ware, C. Pontbriand, and M. Tivey, “An integrated, underwater optical/acoustic communications system,” in OCEANS’10 IEEE SYDNEY, 2010, pp. 1–6.
