ADVANCED AUTONOMY FOR UUV-BASED SYNTHETIC APERTURE SONAR

BD Todd Naval Surface Warfare Center, Panama City, Florida, USA
II Rodriguez-Pinto Naval Surface Warfare Center, Panama City, Florida, USA
DD Sternlicht Naval Surface Warfare Center, Panama City, Florida, USA
J Weaver Naval Surface Warfare Center, Panama City, Florida, USA
M Hartzog Naval Surface Warfare Center, Panama City, Florida, USA

1 INTRODUCTION AND BACKGROUND

The forefront of unmanned underwater technologies is currently centered on mission specific advanced autonomy. Recent focus has been on fast, high fidelity remote sensing to obtain situational awareness and environmental information for maximizing mission success in the maritime domain. In addition, efforts have been made on removing the “operator-in-the-loop” to increase safety, reliability, and speed of operations. Complete architectures for performing autonomous surveys have been developed for integration into modern unmanned underwater vehicles (UUVs) that can rapidly execute pre-defined mission plans (1)**.** These architectures also provide the ability for advanced behaviors by generating tasks and performing actions during the course of its mission (2). In order for an autonomous system to perform advanced behavior, knowledge of its operating environment is crucial, which requires the ability to both sense its immediate environment and extract pertinent information from each environmental measurement that can then be acted upon. A complete unmanned system capable of advanced autonomy would therefore need to be outfitted with onboard capabilities for various data measurement collections, processing, and behavioral rules to accurately determine and respond to relevant information.

UUVs can be equipped with a variety of sensor technologies dependent on application, including acoustic, electromagnetic, and optical sensors. These sensors collect measurements about the UUV’s immediate environment and act as the first step towards advanced autonomy behaviors. Collected measurements are commonly post processed into higher fidelity and/or easier to interpret forms after collection. For many undersea applications, sonar systems are used to obtain long range, high-resolution information about the undersea environment that optical and electromagnetic sensors cannot provide. In missions attempting to observe the seafloor, acoustic imaging is a standard method of obtaining such long-range, and therefore high area coverage, high-resolution information. Synthetic Aperture Sonar (SAS) has been the focus of the majority of recent efforts due to its capability for generating high-resolution imagery of clearly visualized seabed anomalies (3,4). SAS takes advantage of the UUV’s motion, producing a large sonar aperture by processing overlapping pings recorded along the vehicle track. Images are generated via time-delay or ωk beamforming. The resultant image contains highlights and shadows representing the target strength of objects situated on the seafloor (5,6).

Computer aided detection (CAD) and classification (CAC) are broad tools that have myriad applications across the civilian and military sectors, and have become critical to operations that necessitate large quantities of sensor information. In conventional undersea operations, after deployment and information collection from a sensor system, a post-mission analysis on collected data would be performed prior to further action. Traditionally, an expert human observer would need to visually inspect each data sample and identify objects of interest (OOI) – a process that requires extensive time and resources to complete. Modern computational analysis tools allow for a drastic reduction in time by utilizing image processing algorithms and machine learning tools to refine the search space for the human operator from the entire dataset to detected/classified objects of interest.

Algorithms that perform this process, referred to as automated object recognition (AOR), encompass state-of-the-art techniques that include decluttering, denoising, object identification, advanced image processing, and more recently deep learning artificial neural networks (ANNs) (7-10). Systems utilizing AOR have been shown to reduce operator workload while retaining accuracy, allowing for expedited mission completion and the potential to improve mission quality. At its core a CAD/CAC system implementing AOR comprises a sequential process of object detection and

classification, where the output information (such as location) on detected/classified objects of interest can be processed and analyzed for further mission-specific actions.

In underwater applications AOR is widely used in conjunction with processing techniques for image based sensors (typically optical or sonar), and has seen rapid advances in the areas of mine countermeasures (MCM) and geological engineering. Approaches using optical imagery include visualizing breakwaters (12) and undersea cables (13). Application of CAD and AOR algorithms to SAS imagery has proven successful for locating mine-like objects (MLO) in various types of seafloor conditions (14-16). Additionally, the monitoring of undersea infrastructure such as pipelines and cables have benefitted from AOR algorithm implementations on unmanned systems that effectively inform further mission planning (17-19). In these cases anomalies that the CAD/CAC systems are attempting to locate are long, linear objects (LLO) (with an occasional hyperbolic/parabolic curvature present) that span a large swath of the image (20,21). Recent efforts employ SAS imagery to detect underwater scars, cables, and pipelines (22). With sequential detections across multiple images, tracking can be performed using Kalman filters, Bayesian approaches, and particle filters (23-25). These recent advancements enable effective information processing that can elicit advanced autonomy behaviors based on the mission plan.

As improvements in embedded hardware continue to increase computational power, the collocation of complex data analysis processes onboard the vehicle becomes more feasible. Unmanned platforms are typically equipped with a suite of different sensors that collect data during autonomous operation. When post-processing and analysis of the multiple data types were required, this work traditionally had to be performed “topside” – which required the data to be copied from the vehicle hardware onto more computationally powerful computers. Recently, embedded platforms have reached a baseline allowing computationally expensive tasks among multiple data streams to be conducted on the vehicle in real-time. The traditional “topside” approach to acquiring identification quality imagery of an object would require a multiple sortie approach, as the initial survey data would need to be returned and processed by a combination of operations and computer aids before a second sortie would be planned and sent to reacquire higher quality imagery of the target. In addition to time and operator concerns, this approach also is susceptible to reacquire error from drift in the inertial navigation system (INS) resulting in incorrect position, a problem that is made more pronounced if the vehicle is unable to acquire a global position system fix.

The purpose of the work described in this paper is to enable advanced autonomous UUV operation capable of performing underwater search, detecting objects of interest, and utilization of detections to elicit a behavioral decision. To this end, an AOR software module was integrated on the embedded hardware of a UUV to perform real time search for multiple objects of interest in SAS imagery. Conceptually, the AOR module fits between the lower-level SAS module (where raw environmental information is collected and processed into imagery) and the higher-level autonomy modules (which control vehicle positioning, motion, and tasking). Successful integration required holistically considering: the nature of sensor-specific data; capabilities of the embedded hardware; and communication between the AOR other key system components. Section 2 of this paper addresses automatic object recognition; Section 3 addresses vehicle autonomy; Section 4 describes operational results; and Section 5 discusses future work and challenges.

2 AUTOMATIC OBJECT RECOGNITION

2.1 GENERAL METHODOLOGY

In order to make behavioral tasking decisions required for a mission, the AOR must be capable of detecting and classifying two main target types: long continuous linear objects and small discrete objects. This capability is achieved by performing two separate AOR operations on the same data sample to search for the different target types. In an ideal implementation, these two AOR streams would run in parallel and the results from both streams would feed into a data fusion process to increase confidence in the detections. However, in the current implementation, the two AOR streams are run consecutively due to hardware processing limitations. Prior to describing the current

implementation of the AOR algorithms, information on the UUV mounted sonar system data products, sonar imagery, and their roles in the UUV application are provided.

The data being collected and processed onboard the vehicle is from an experimental dual-band synthetic aperture sonar mounted on a medium-sized UUV. The operational concept of this project requires that the AOR be capable of finding both long continuous objects and small discrete objects lying on or near a continuous object. Finding these two different types of objects necessitates two separate AOR processes, because the objects are distinct enough in terms of shape and appearance in the returned sonar signal that a single AOR is unsuitable to find both. The structure of the AOR for this work, therefore, is a parallel set of two AOR algorithm pipelines. When a new sonar image arrives from the system after being beamformed into a SAS image, the image is independently processed by each separate AOR pipeline. In the current implementation, these steps are not truly performed in parallel due to hardware and software constraints, but in general, these are fully independent AOR processes, which should allow for simultaneous computation.

The AOR pipelines for both continuous and discrete objects follow the typical processing stream for data within the Modular Algorithm Testbed Suite (MATS) software developed at the Naval Surface Warfare Center, Panama City Division (NSWC PCD). The processing stream was originally developed to recognize discrete objects in sonar images, but the steps of this process can be abstractly applied to any recognition operation:

1. Normalize – this step takes the input data (*i.e.,* an image) and produces a data object of the same type (*i.e.,* a normalized image) by applying data processing (*e.g.,* smoothing, scaling, clipping, cropping) to the *entire* input data.
2. Detect – this step takes a set of data (*i.e*., the normalized image from the previous step) and applies a prescreening operation that identifies potential regions of interest in the data. Each potential region of interest is designated a *contact*, and the detection step serves as a low-cost method to downsample the input data to a list of contacts, which are the output of this step.
3. Extract Features – computations are applied to the subset of the data in the region of interest. The feature computation can be any operation (or collection of operations) that maps from the data contained in a contact's region of interest to a vector space called *feature space*. The vector representing a particular contact in feature space is called the contact's *feature vector*, and is the output of this step. A feature space is typically considered good if the subspace of feature vectors belonging to true positive contacts are well-separated from the subspace of feature vectors belonging to false positive contacts.
4. Classify – this step takes the contact feature vectors as input, and classifies them as objects of interest (or not) based on a decision boundary in feature space. This decision boundary must be formed using a set of training examples. The output of the classification step is a set of predictive labels identifying each contact as an OOI or non-OOI.

We note that not all recognition methods necessarily use all of these steps and there may be additional steps depending on application. For instance, identification may serve as a step beyond classification and is typically performed by a human operator. Compared to classification, which supports determination of whether the assumedly manmade object is of interest or merely a distraction, identification would be a step toward determining the accuracy of that classification and what type of object the contact is. For skipping steps, a convolutional neural network (CNN) trained for recognition might skip the detection step, allowing every element of the input data to be a contact. Furthermore, the early layers of the CNN would represent a step similar to the feature extraction process, while the last layers serve a purpose similar to classification. In our work, the continuous object detector lacks the feature extraction step, and the classification process is more focused on consolidating contacts rather than accepting or rejecting them.

2.2 DISCRETE OBJECT DETECTOR

This section outlines the processing applied to each SAS image sent into the AOR module. The steps are described in the order in which they are applied. The input data is a raster of 32-bit complex-valued data stored in an HDF5-formatted binary file. This file includes additional information, such as navigation positions of the sensor and resolution of the raster pixels.

The normalizer module performs pre-processing actions on the image data required to improve the AOR process. These include image-wide operations such as time-varying gain (TVG),

smoothing, de-noising, and cropping. The precise normalization processing varies from sensor to sensor, as the characteristic properties of the images they produce can vary widely. For the SAS sensor, the MATS normalization applies a time-varying gain to adjust the image background energy level to be roughly uniform with respect to range. It is the job of the normalizer to remove these artifacts by cropping the image to the optimized image segment, the portion of the range that lies between the nadir gap and the end of the effective range, in which the imagery is of sufficiently high quality to continue with AOR processing.

The anomaly detector is computationally efficient so that it can be applied to the entire image with low computational cost. Detected contacts are stored in a list, and the subsequent AOR steps are performed only on the contacts and their associated snippets. Anomaly detection is performed with a modified Reed-Xiaoli (MRX) filter for blob detection (26). This detector determines whether a blob-like anomaly is located within the bounds of the core filter by comparing the statistics of the pixel intensities within the core filter to the statistics of the pixel intensities within the ring filter. While the discrete OOIs for this work are somewhat different from the objects that the MRX detector was originally designed for, it is sufficiently general such that our OOIs are flagged and added to the contact list. The classification step applies labels to the contacts produced by the anomaly detector. Classification proceeds by mapping the contacts to a feature space, and applying a decision metric directly within the feature space. In this context, we are concerned with binary classification, which predicts only whether each contact is an OOI or not. Our first approach has been to simply use a classifier that was trained to find small objects in SAS data, and to judge later whether this classifier is able to separate our discrete objects of interest from clutter. The classifier chosen for this purpose uses a relevance vector machine (RVM) to construct a decision boundary in the feature space. The feature vector of each contact is comprised of the Fourier coefficients of the image snippet and the response of the snippet to the Haar wavelet decomposition.

2.3 CONTINUOUS OBJECT DETECTOR

The detector module for the continuous object AOR consists of an image processing and pattern recognition pipeline that operates with high computational efficiency. The detector utilizes a combination of SAS imagery, taken at both the high frequency and low frequency bands of the SAS system, and outputs a contact list containing the locations of detected LLOs. The AOR for continuous objects was developed to find LLOs that span, or nearly span the entire sonar image. In contrast to the detector for discrete objects, the continuous object AOR does not extract snippets of each contact, requiring the processing steps be applied on the entire image for each frequency band. The final contact list consists of a set of georeferenced points describing the location of the LLO, as well as the bearing. For each SAS image in their respective frequency bands, the normalization module is first invoked, for data smoothing and cropping to the “sweet spot.” Then the detection process starts with a tone mapping operator, an algorithm designed to optimally map the dynamic range of the 32-bit image raster to an 8-bit image raster for faster processing (27). The image is then denoised using Gaussian and median filters to decrease the influence of speckle noise on the ability to extract edges in the image. An edge map for each image is then obtained via an adaptive, gradient-based edge extraction routine, based on the method derived by Canny (28). The edge extraction method first computes the gradient intensity representation of the image, and subsequently applies non-maximum suppression to remove “false” responses to the edge detection. The removal of these false responses is equivalent to thinning the edges, functioning by removing pixels at lower intensity values along each edge. Hysteresis thresholding is then performed to enforce edge definition criteria, by suppressing “weak” edges that are not connected to “strong” edges. Edges below the lower intensity threshold boundary are removed and edges above the upper intensity threshold boundary are retained. Edges located in between threshold settings are retained if connected to another retained edge. Lower and upper intensity thresholds are adaptively computed from the median intensity value of the image, with the threshold “tightness” computed at 45% above and below the median. With the edge map extracted, pixels corresponding to the LLO and image noise are further filtered using morphological operations. After a series of morphological operations including dilation, opening, and closing, the final edge map is passed to a Progressive Probabilistic Hough Transform (PPHT), which finds and estimates line segments (29). The set of line segments detected by the PPHT largely outlines the LLO, but has a tendency to generate overlapping line segments of varying lengths. In

addition, noise from the edge map may induce false line segment detections that must be filtered further to accurately detect the LLO. To further refine the detection to correctly find a LLO, the pruning/characterization steps are performed on the output of the PPHT. First, each line segment from the PPHT is converted to a series of endpoints, which are then clustered using density based spatial clustering of applications with noise (DBSCAN) (30). The DBSCAN algorithm clusters the sets of endpoints to consolidate multiple overlapping detections into one detected LLO segment. If the LLO is curved or bent, the PPHT typically detects shorter straight line segments along the curve. Then, the clustering algorithm results in a set of line segments that describe the LLO. In these cases, contact correlation is performed, where each line segment is pairwise compared for sequential linearity. The sequential linear line segments are then implemented into a spline fitting algorithm to extract the final contact.

The processing pipeline occurs for each SAS image at their respective frequency band. After the PPHT step in the pipeline, line segment detections are obtained from each image, to then act as the input for the pruning/characterization step. The structure of a LLO can create difficulties for the high frequency band of the system. In many cases, LLOs with small diameters can be difficult to detect in a SAS image, even by eye. The seafloor complexity, bottom type, and potential for burial all contribute to the difficulty in visualizing the LLO in the high band. However, images generated by the low frequency band can compensate for the limiting factors, depending on the material properties of the LLO. The low frequency band was primarily used during the project for its capability in finding small diameter LLOs. This is due to both the enhanced LLO response and sediment penetration at the lower frequency, helping to resolve partially buried LLOs. One aspect of the low frequency imagery used to advantage is that the water/sediment interface can be treated largely as noise, allowing the LLO to stand out. Ultimately, fusing information from both the high and low bands will be used to assess the physical state of an LLO, and flag the presence of structural or environmental anomalies.

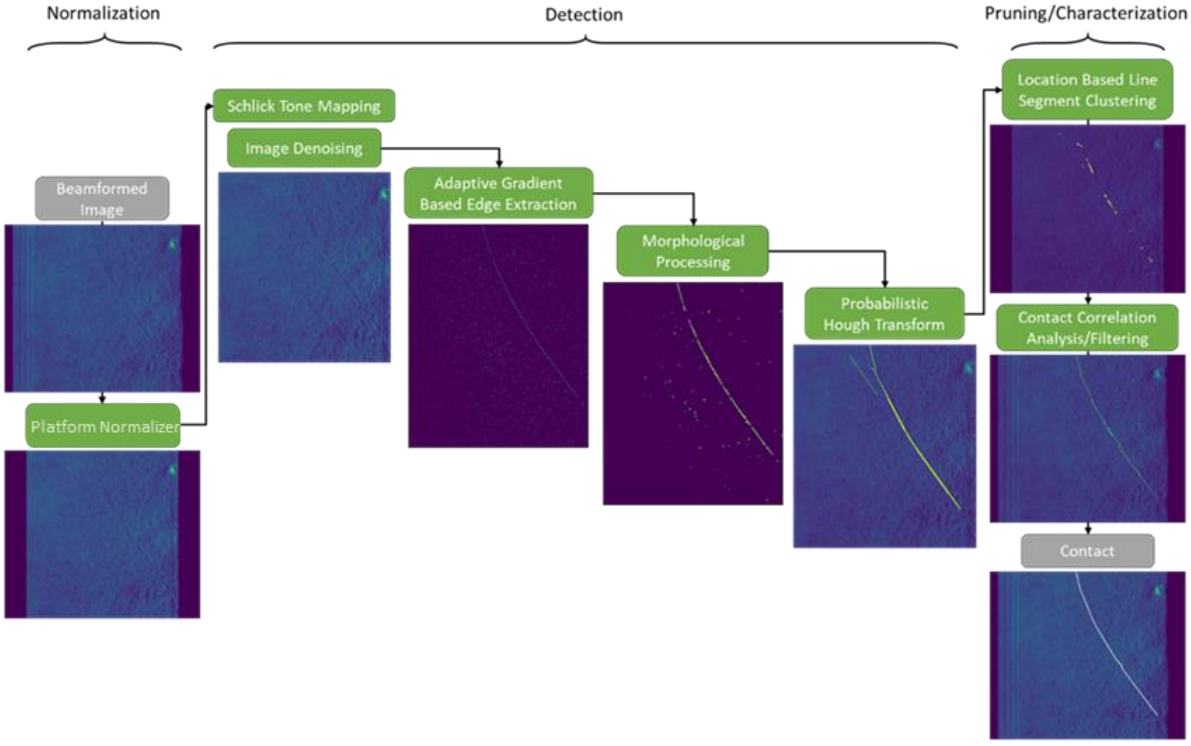

Figure 1: Image processing pipeline for LLO detection and clustering.

3 AUTONOMY

3.1 AVA OVERVIEW

As introduced earlier, a modular autonomy architecture was developed by NSWC PCD to support S&T demonstrations and testing of sensor capabilities (1). The Autonomous Vehicle Architecture (AVA) provides a framework to allow for rapid integration of new capabilities. Base interfaces throughout the architecture provide general functionality and connections required between each element, allowing new tasks and behaviours to interact automatically. The core of AVA, shown in Figure 2, consists of three layers: mission and sortie management, deliberative task planning, and low-level reactive behaviors. Replanning and perception components operate in parallel with the core framework, providing information to elements within the architecture or allowing modification of the current plan the autonomy is following. Interfaces are provided within the replanning framework that allow the creation of monitors and solvers. Monitors scan and track discrepancies within the environment, such as faults with sensors or new objects of interest that are not handled by a running task. Solvers are components that register for specific discrepancies and create a plan that solves the discrepancy by creating new tasks, modifying parameters, or other required changes within AVA.

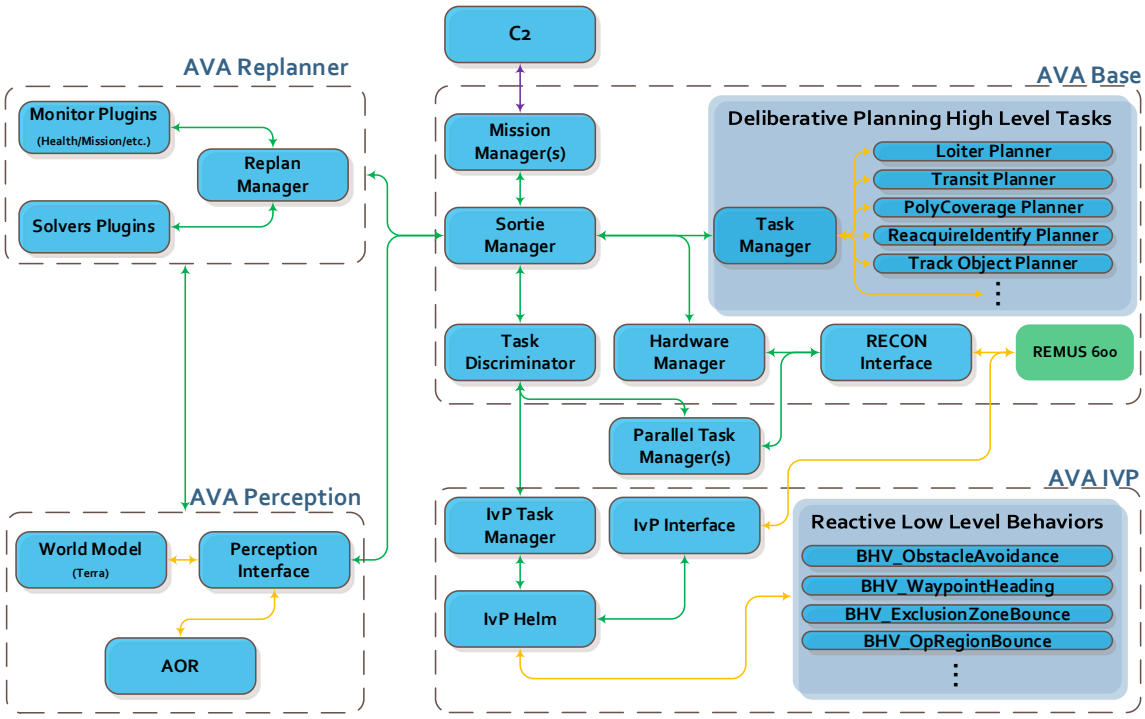

Figure 2: Components and flow of the Autonomous Vehicle Architecture.

The architecture is built on top of the Robot Operating System (ROS) 2.0 environment for intercommunication between elements. Additionally, a low-level behaviour planning toolset from the Mission Oriented Operating Suite Interval Programming (MOOS-IvP) environment has been converted to ROS and operates as one of the potential reactive layers for AVA. With these software environments, AVA is able to follow the Modular Open Software Approach (MOSA), allowing components to be swapped in or out of the framework or moved to other architectures as needed. AVA was also written to be platform and domain agnostic, being demonstrated on multiple undersea, surface, and ground systems.

3.2 Replanning and Tracking Autonomy

To achieve the goal of tracking a continuous LLO and identifying discrete objects of interest, four elements were created within AVA: (1) an AOR monitor which senses when continuous or discrete objects are detected; (2) a track continuous object solver which creates an execution replan for tracking continuous objects; (3) a discrete object solver which creates an execution replan to identify discrete objects; and (4) a tracking continuous task which is prompted to create a waypoint plan when the tracking continuous solver includes both a track task in its replan and a continuous object is received. A discrete identifying task already existed within AVA and was used during testing of these capabilities.

To demonstrate each of these functionalities, a simple survey is started at a given location and each of these components listed above are enabled. When the vehicle is surveying a region and a continuous object is detected, the monitor will flag a discrepancy notifying the track continuous solver that a solution is required. Once a replan solution is accepted, the track continuous task will be prompted to provide a waypoint list for execution. Using the contact parameters provided by the AOR, a waypoint list is generated and executed. During track task execution, one of three execution pathways might be taken depending on what the AOR is sensing and providing the autonomy.

The first is that no discrete objects will be detected, and no more continuous objects will be detected. In this scenario, the vehicle will return to a survey task which attempts to find more continuous objects to restart tracking on a new object of interest. The second scenario is one in which no discrete objects are found during a tracking task, but more continuous objects are detected. Because in this scenario the task executing is a tracking task, no replan components are required and the new contact is processed immediately by the tracking algorithm and the new waypoints generated are immediately executed. The third and last scenario is one in which discrete objects are received while executing a tracking task. In this scenario, the discrete object solver will create a replan that will prompt an identification task algorithm to create a waypoint plan to identify the discrete object or objects. At this point, the vehicle will save its current position on the track task and it will break off and execute the identification task. After the identification, the vehicle will return to its saved position and a new track task from that point will be executed in order to complete the remaining portion of the tracking task. From this point any of the previous scenarios (including this one) can be triggered and executed. These execution pathways allow the vehicle to survey regions for objects of interest, or indefinitely track a continuous object while identifying discrete objects along the way.

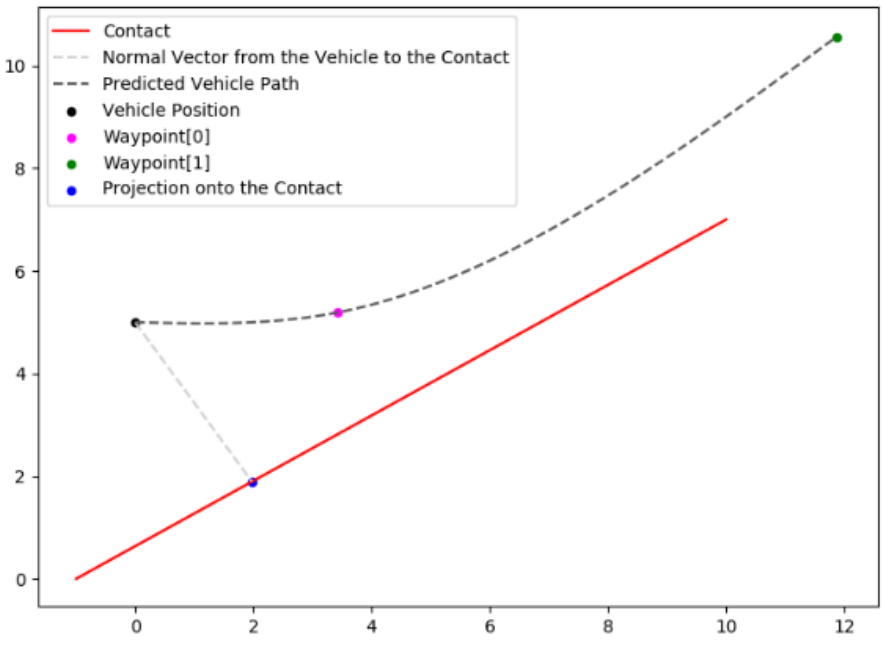

Figure 3: Depiction of a track behaviour scenario.

The Track Continuous task contains an algorithm that performs a series of steps to produce waypoint goals for a vehicle to navigate a parallel track alongside an object at a fixed (but configurable) distance, as seen in Figure 3. The algorithm can produce points such that an optimized path can be achieved by an unmanned system of varying vehicle dynamics. It also intelligently produces goals that promote the vehicle to continue its track in its current forward direction. It also ensures it tracks on the side of the contact in which the vehicle is already positioned – which is usually the side in which the contact was first perceived. As a task is being executed, new contacts can be received and processed. This results in an updated waypoint list, based on a more relevant and recent contacts.

4 RESULTS

4.1 DETECTOR RESULTS

Figure 4 shows a collection of SAS contact snippets identified as anomalies by the MRX detector. These images demonstrate that exceptionally bright (that is, noticeably brighter than background alone) sonar returns concentrated in a small area are the key characteristic for identifying an anomaly, rather than shape or other more specialized features.

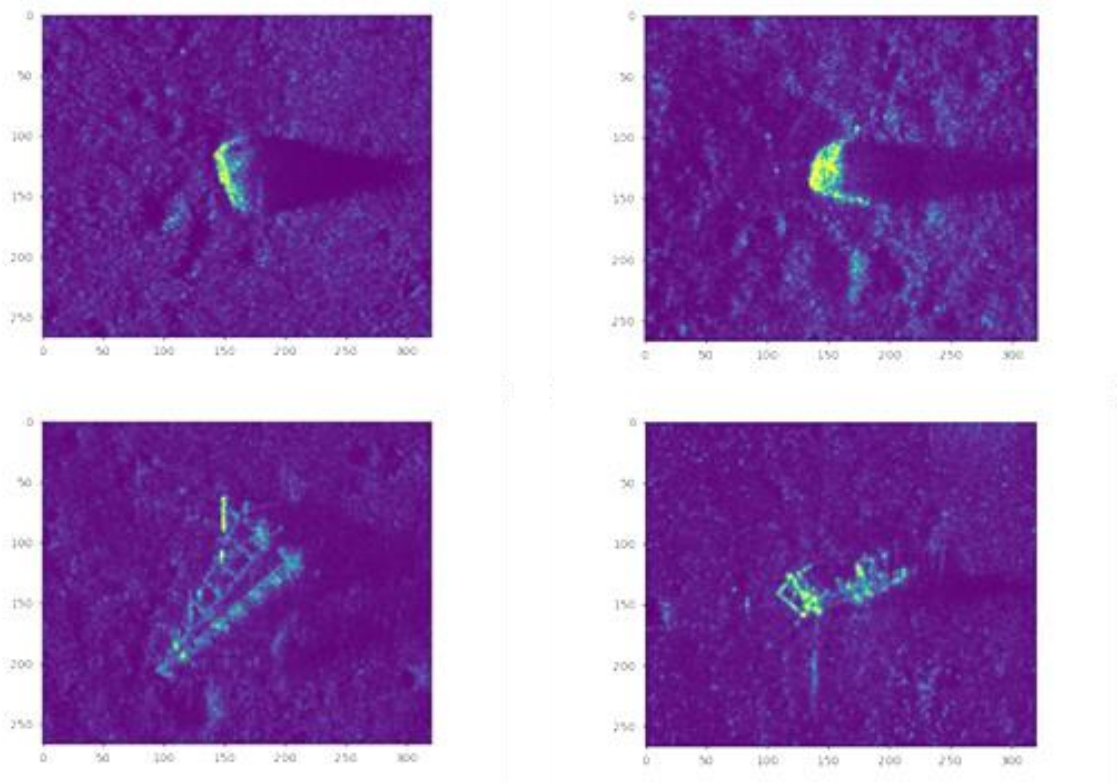

Figure 4: Example MRX detections in SAS imagery. Axes represent pixel sample numbers, which are then mapped to geographic coordinate space.

The continuous detector was tested in multiple environmental and situational conditions to assess robustness and reliability. Environmental conditions, such as the presence of detritus, can occlude a section of a LLO, increasing difficulty of detection. Results have shown detection robustness against the presence of occlusions, with no spurious detections induced by occluding detritus. Another condition tested was a heterogeneous seafloor, with rocks of various size that act as clutter and noise. Results for this case yielded detection robustness, where rock clutter is filtered out in the processing pipeline. The combination of two environmental conditions was also tested, where the LLO was present within an area with heavy clutter while also being occluded by small rocks and shadows from

larger rocks. Even in this case, the detection algorithm successfully detected the LLO and obtained its location. Finally, a two LLO situational condition was investigated, assessing whether the detection algorithm was capable of detecting two individual LLOs within the same image. The detection pipeline was capable of isolating both LLOs, returning two contacts with accurate locations.

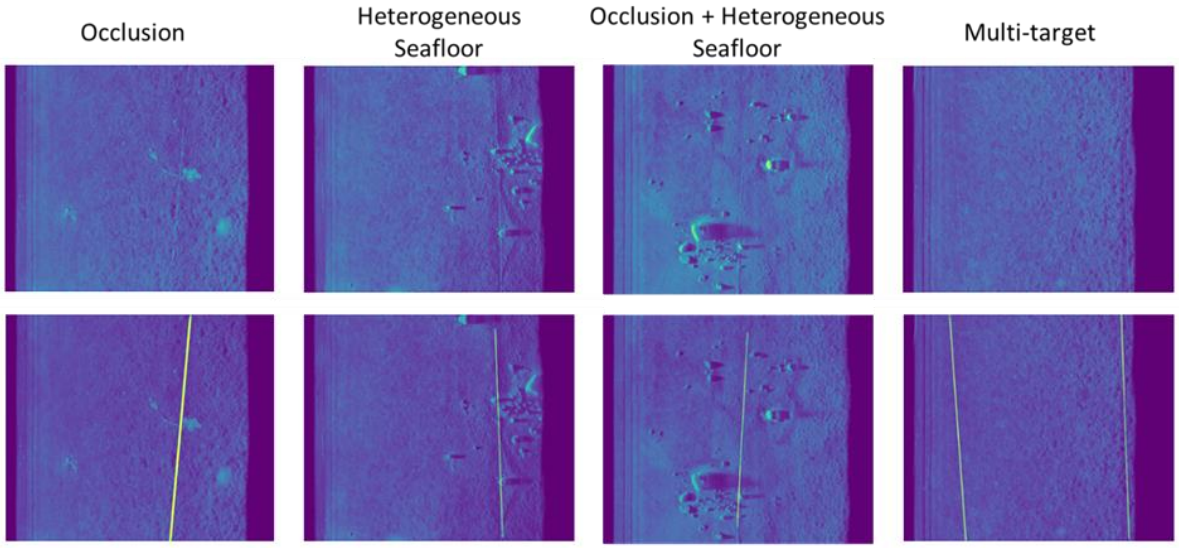

Figure 5: Example LLO detections in SAS imagery.

4.2 FULL OPERATION RESULTS

The data used for developing and testing the AOR algorithms was collected from 2021-2022 in a coastal bay situated in the northwestern Atlantic ocean. The area contained a LLO situated on a heterogeneous silt/mud bottom containing rocks and detritus at ~10m depth. The UUV platform traversed at altitude 5m above the seafloor and the LLO was insonified and imaged using onboard SAS processing. The SAS imagery was processed by the onboard AOR algorithms to produce contacts that were used to perform autonomy redirects for tracking and imaging.

The accomplished goal of the operational test was to perform a full autonomy mission using onboard AOR queues through the steps of: i) Ingress, ii) Survey, iii) Track LLO, iv) Image Object near LLO, and v) Egress with collected imagery. Imaging of the object near the LLO was performed using Circular Synthetic Aperture Sonar (CSAS), a higher resolution SAS imaging technique (31). In the run shown in Figure 6, the UUV ingressed into the survey area and began a lawnmower search pattern with 600m long legs at a spacing of 60m. After entering into the search pattern, the onboard AOR detected the presence of an LLO and an autonomy replanning event was triggered, instructing the UUV to follow the LLO using a forward projection of the LLO's found position. After tracking the LLO for several consecutive images an object of interest was identified in a contact from the discrete AOR, whereupon the UUV stopped tracking and began capturing a CSAS image of the object. After capturing the imagery, the UUV returned to an egress point designated for imagery pass-off to an operator – though in other performance configurations the UUV could have returned to the queued tracking of the LLO, finished mapping, and initiated search for other objects of interest.

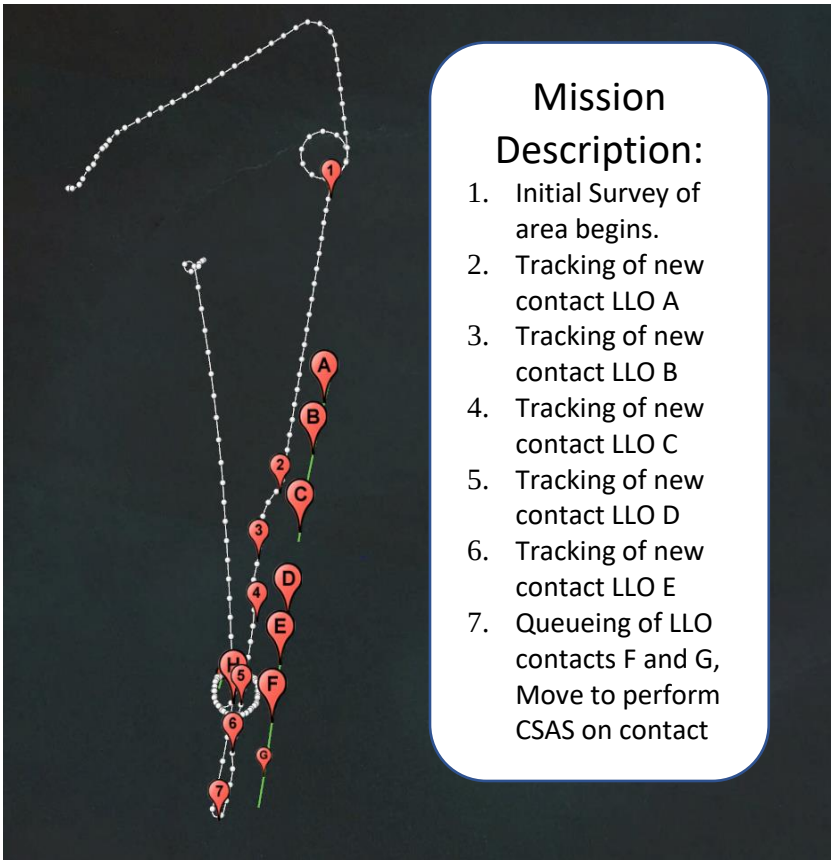

Figure 6: Example of a full mission run with advanced reactive autonomy.

5 DISCUSSION AND FUTURE WORK

This effort successfully developed an integration schema for a robust sensing-to-behavior pipeline. Various subsystems, including a SAS sensor, an AOR engine, and an autonomy architecture were integrated with proper networked communications protocols to enable environmental information-based advanced autonomy on a UUV. In addition to further refinement of the communications between modules, future effort is needed to allow tailoring AOR algorithms for mission needs. The main thrust of future work is in the development of predictive classification models based on collected data of our given objects of interest. Currently, this effort is challenged by the modest amount of labeled OOI data available. For our small set of training data, transfer learning is likely the most suitable approach, using a version of the resnet classifier that has been trained to find small objects in SAS data extracted by the MRX detector. The transfer learning process would initialize the model, and then train with new data for a small number of epochs (3-5 epochs). Current results indicate that this model is still not suitable for reliable classification and requires a significant increase in training data on OOIs in order to obtain a better predictive classification model that could differentiate between a large variety of object types.

For an unmanned system to successively sense, perceive, and take action, determining what comprises *actionable information* is necessary. Future work is needed to define actionable information in the context of the sensing-to-behavior pipeline – a research area that is critical to further developing advanced autonomy for UUV-based synthetic aperture sonar.

6 ACKNOWLEDGEMENT

The authors gratefully acknowledge guidance and support from NSWC PCD Command and S&T Department Leadership, and the NAVSEA 219 Innovative Science & Engineering (NISE) program.

7 REFERENCES

1. Weaver, J., J. Perkins, and D. Sternlicht. “Advanced Autonomy Architecture for Maritime Applications AVA,” in *OCEANS 2016 MTS/IEEE, Monterey*, Monterey, CA, USA, 2016, pp. 1-8, doi: 10.1109/OCEANS.2016.7761389.
2. Sternlicht, D., et al. “Advanced Sonar Technologies for High Clearance Rate Mine Countermeasures” in *OCEANS 2016 MTS/IEEE Monterey,* Monterey, CA, 2016, pp. 1-8, doi: 10.1109/OCEANS.2016.7761133.
3. Sternlicht, D., and J. Pesaturo, “Synthetic Aperture Sonar: Frontiers in Underwater Imaging.” *Sea Technology*, vol 45, no. 11, 2004, pp. 27-34.
4. Cobb, J., et al. *A Method to Simulate Synthetic Aperture Sonar Images with Parameterized Autocorrelation Functions*, SPIE, 2010, pp. 76640Y-76640Y, doi:10.1117/12.851110.
5. Hansen, R., “Synthetic Aperture Sonar Technology Review.” *Marine Technology Society Journal*, vol. 47, no. 5, 2013, pp. 117-127.
6. Sternlicht, D., et al. “Advanced Sonar Technologies for Autonomous Mine Countermeasures,” in *OCEANS'11 MTS/IEEE KONA*, Waikoloa, HI, 2011, pp. 1-5, doi: 10.23919/OCEANS.2011.6107149.
7. Bhanu, B., and T. Jones, “Image Understanding Research for Automatic Target Recognition,” in *IEEE Aerospace and Electronic Systems Magazine*, vol 8, no.10, 1993, pp. 15-23, doi: 10.1109/62.240102.
8. Dobeck G., et al. *Adaptive Large-Scale Clutter Removal from Imagery with Application to High-Resolution Sonar Imagery,* SPIE, 2010, pp. 76640X–76640X.
9. Isaacs, J. C., “Sonar Automatic Target Recognition for Underwater UXO Remediation,” in *2015 IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, Boston, MA, 2015, pp. 134-140, doi: 10.1109/CVPRW.2015.7301307.
10. Tian, Y., L. Lan, and H. Guo, “A Review on the Wavelet Methods for Sonar Image Segmentation.” *International Journal of Advanced Robotic Systems*, vol. 17, no. 4, SAGE Publishing, 2020, p. 172988142093609.
11. Neupane, D., and J. Seok,”A Review on Deep Learning-Based Approaches for Automatic Sonar Target Recognition.” *Electronics*, vol. 9, no. 11, MDPI, 2020, p. 1972.
12. Chen, H., W Chuang, and C. Wang, “Vision-based Line Detection for Underwater Inspection of Breakwater Construction Using an ROV.” *Ocean Engineering*, vol. 109, Elsevier BV, 2015, pp. 20-33, https://doi.org/10.1016/j.oceaneng.2015.09.007.
13. Ortiz, A., J. Antich, and G. Oliver, “A Bayesian Approach for Tracking Undersea Narrow Telecommunication Cables,” in *OCEANS 2009-EUROPE,* Bremen, Germany, 2009, pp. 1-10, doi: 10.1109/OCEANSE.2009.5278108.
14. Hożyń, S., “A Review of Underwater Mine Detection and Classification in Sonar Imagery.” *Electronics*, 2021, p. 2943, https://doi.org/10.3390/electronics10232943.
15. Stack, J. “Automation for Underwater Mine Recognition: Current Trends and Future Strategy.” *Detection and Sensing of Mines, Explosive Objects, and Obscured Targets XVI*. Vol. 8017. SPIE, 2011, pp. 205-225.
16. Williams, D., “Fast Target Detection in Synthetic Aperture Sonar Imagery: A New Algorithm and Large-Scale Performance Analysis,” in *IEEE Journal of Oceanic Engineering*, vol. 40, no. 1, 2015, pp. 71-92, doi: 10.1109/JOE.2013.2294532.
17. Kojima, J., “Cable Tracking by Autonomous Underwater Vehicle,” in *2003 International Conference Physics and Control. Proceedings (Cat. No. 03EX708)*, Tokyo, Japan, 2003, pp. 171-174, doi: 10.1109/SSC.2003.1224135.
18. Szyrowski, T., et al. “Subsea Cable Tracking by an Unmanned Surface Vehicle.” *Underwater Technology*, vol. 32, no. 4, 2015, pp. 217-229.
19. Villar, S., et al. “Evaluation of an Efficient Approach for Target Tracking from Acoustic Imagery for the Perception System of an Autonomous Underwater Vehicle.” *International Journal of Advanced Robotic Systems*, vol. 11, no. 2, 2014, p. 24.
20. Xu, C., et al. “Review of Underwater Cable Shape Detection.” *Journal of Atmospheric and Oceanic Technology*, vol. 33, no. 3, 2016, pp. 597-606.

21. Szyrowski, T., et al. “Developments in Subsea Power and Telecommunication Cables Detection: Part 1-Visual and Hydroacoustic Tracking.” *Underwater Technology*, vol. 31, no. 3, 2013.
22. Isaacs, J. C., and R. Goroshin, “Automated Cable Tracking in Sonar Imagery,” in *OCEANS 2010 MTS/IEEE SEATTLE*, Seattle, WA, 2010, pp. 1-7, doi: 10.1109/OCEANS.2010.5664414.
23. Asif, M., and M. Arshad. An Active Contour and Kalman Filter for Underwater Target Tracking and Navigation. INTECH Open Access Publisher. 2006.
24. Ortiz, A., M. Simó, and G. Oliver. “A Vision System for an Underwater Cable Tracker.” *Machine Vision and Applications*, vol. 13, 2002, pp. 192-140.
25. Szyrowski, T., et al. “Subsea Cable Tracking in an Uncertain Environment using Particle Filters.” *Journal of Marine Engineering & Technology*, vol. 14, no. 1, 2015, pp. 19-31.
26. Reed, I.S., and X. Yu. “Adaptive Multiple-Band CFAR Detection of an Optical Pattern with Unknown Spectral Distribution.” *IEEE Transactions on Acoustics, Speech, and Signal Processing* 38, no. 10 (October 1990): 1760–70.
27. Schlick, C.,. “Quantization Techniques for Visualization of High Dynamic Range Pictures.” *Photorealistic Rendering Techniques*, vol. 20, 1994.
28. Canny, J., “A Computational Approach to Edge Detection” in *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. PAMI-8, no. 6, 1986, pp. 679-698, doi: 10.1109/TPAMI.1986.4767851.
29. Kiryati, N., et al. “A Probabilistic Hough Transform.” *Pattern Recognition*, vol. 24, no. 4, 1991, pp. 303-316.
30. Ester, M., et al. “A Density-Based Algorithm for Discovering Clusters in Large Spatial Databases with Noise.” *Knowledge Discovery and Data Mining*, 1996, pp. 226-231.
31. Sternlicht, D., J. Fernandez, and T. Marston, “Detecting, Classifying Mines with Synthetic Aperture Acoustic Tomography.” *Sea Technology*, vol 53, no. 12, 2012, pp. 10-13.