A Study of Image Processing Applications in Autonomous Marine Vehicles using Forward-Looking Sonar

Sourajyoti Datta, *MSc, Technische Universität Kaiserslautern,*

***Abstract*—The exploration of the submarine world using Autonomous Marine Vehicles (AUV) is growing and has numerous applications. The autonomous long-term, safe and reliable traversal of these AUVs is a critical ability needed for such applications. In that aspect, physical interpretation and semantic understanding of the dynamic environment of an AUV using imaging sensors of the vehicle is of paramount importance. This study explores the limitations of underwater optical imaging along with acoustic devices such as Sonars that are employed to alleviate the issues, and various Image Processing techniques, including Machine Learning and Deep Learning methods, applied on the acoustic imagery for applications such as object detection, classification, tracking, and avoidance of obstacles. Furthermore, the practical feasibility of these techniques accounting for the ability to be executed in real-time efficiently is studied.**

***Index Terms*—Autonomous Marine Vehicle, Forward-Looking Sonar, Image Processing, Acoustic Image Processing, Machine Learning, Deep Learning**

I. Introduction

In recent years, the use of Autonomous Marine Vehicles for navigation and exploration has been growing. Alongwith the numerous improvements in the vehicles themselves as a tool for various applications, the incorporation of useful autonomous capabilities in these vehicles is also an ongoing effort. Since such autonomous functions can vary immensely, attributing to different modules of the vehicles, a specific focus on the capabilities based on the processing of imaging data generated using acoustic sensors, and their application thereof, is under study here.

One primal autonomous function of AUVs relates to understanding the physical environment surrounding the vehicle, which allows for different tasks. These tasks range from creating a map of the immediate local environment to detecting and identifying objects in its vicinity, tracking objects, and avoiding obstacles. Furthermore, in the absence of GPS signals underwater, localization and alignment of the vehicle using environmental features are enabled [4]. These tasks, in turn, provide the AUV with long term autonomy encompassing safe, reliable, and robust maneuvering capabilities [3][4]. However, many of these tasks that incorporate processing of imaging data are often computationally heavy, thus adding the real-time executability of these tasks as a fundamental requirement for the practical applicability. In this study, we focus on these particular aspects of the autonomy of an AUV.

S. Datta is with the Department of Computer Science, TU Kaiserslautern, Rhineland Palatinate, 67663 Germany e-mail: datta@rhrk.uni-kl.de (see https://www.linkedin.com/in/sourajyoti-datta/).

Contrary to aerial and ground vehicles that use optical sensors providing RGB images as visual information, the use of sonars producing acoustic images is more suitable for underwater conditions. Due to poor underwater optical visibility as an effect of water turbidity and distortion created thereof, the perception of optical sensors are heavily constrained [5]. Sonars, however, are acoustic sensors that offer the advantage of being invariant to water turbidity [3], hence providing a useful alternative for working under these challenging conditions [5]. In water, based on their frequency, acoustic waves can travel greater distances with smaller attenuation. However, these sonar generated acoustic images still suffer from distortion and noise to an extent, which poses various challenges in processing the data [3][7]. Numerous research studies have shown the feasibility of applying various traditional image processing techniques on these acoustic images in tasks such as object detection, image registration, object association and tracking, obstacle avoidance, navigation, localization, and mapping. Furthermore, the application of advanced Machine Learning (ML) and Deep Learning (DL) techniques on tasks, such as detection and classification of objects, has been successfully demonstrated by various researches.

Most, if not all, of these techniques also demonstrate a methodical workflow to an extent. The visual data generated from the sonars are processed, and necessary information extracted at each task is passed on to the next to perform another set of tasks. For example, information about objects detected in a sonar image by a detection system can then be passed on to an object classification system or an image registration and tracking mechanism to perform these related tasks. Hence, it can be determined that in an effective autonomous system, there could exist a tight coupling of the various tasks undertaken.

The subsequent sections of this paper are structured as follows. In Sec. II, an overarching view of the entire methodical workflow is discussed. Each step, along with the flow of information from one step to the next, is briefed. In the subsequent Sections (III, IV, V, VI, VII), each of the five main segments of the workflow are discussed in detail. Finally, in Sec. VIII, the conclusions drawn from the various empirical studies evaluated are presented. The opportunities and ideas related to future improvement or modification of the methodology are further discussed as well.

II. Methodology

THE methodical workflow, as mentioned before, can be constructed by combining multiple tasks depending on the necessity and the goal of the tasks to be performed. One such workflow, as explored here, can be built using five broad steps. The workflow is outlined visually in Fig. 1, and the consecutive steps are as follows:

- *Sonar Imaging*: The workflow begins with collecting visual data from the sonar. With every ping of the sonar, the acoustic image is collected and sent downstream for further processing. Every successive frame generated is maintained chronologically, which is crucial to some of the methodology's downstream phases.
- *Object Detection*: The next step in the process is the detection of the object(s) in the image. Various Image Processing and heuristic techniques are employed, usually applied per frame of the sonar imagery data, to identify and extract relevant information from the image regarding objects, their location, and their size.
- *Object Classification*: Once the objects are detected, their features are extracted to classify the object. For example, various kinds of objects, like a ship, a ship's wake, stationary objects on the surface or underwater, etc., can be detected and classified to identify them into categories as required. This can provide further heuristics for the association of objects and reduce the number of objects that need to be considered for tracking to a certain extent, thus providing some performance gain in the subsequent steps.
- *Object Association and Tracking*: Once objects have been detected and classified, the objects need to be associated between every consecutive frame(s) of the sonar pings, such that for each unique object, its location in every image frame is known. This enables tracking of the movement of an object over the chronologically arranged sonar pings, either from the AUV's frame of reference or even using keypoints on the global reference.
- *Obstacle Detection and Avoidance*: Once the objects have been associated and tracked, the information is used to identify which of the objects are potential obstacles. The objects' trajectories are computed with respect to the AUV's motion to detect the possibility of collision, hence maneuver the AUV accordingly for the avoidance of the obstacle.

III. Sonar Imaging

SONARS, specifically active sonars, are devices that generate acoustic waves, also called pings, which are emitted outward from the device. These acoustic waves travel through the environment, and once they hit any obstacle, some part of the wave is reflected. The reflected part that returns to the Sonar's sensor array is recorded to form the acoustic imaging data. However, these acoustic waves can also suffer from absorption by the medium itself that the wave is propagating through or by the obstacle it hits, and thus can cause distortions in the image generated.

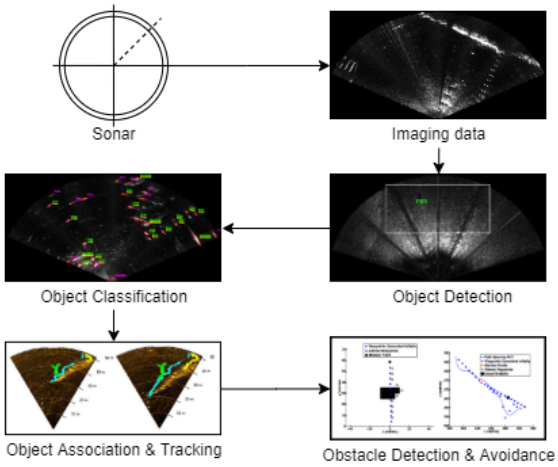


Fig. 1: Workflow of the pipeline. Information flows sequentially from one stage to the next.

A. Sonars

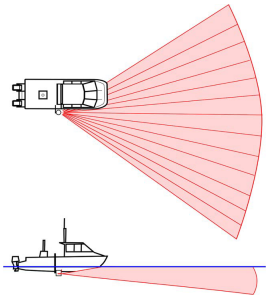
Researchers have often employed various types of sonars, like Profiling Sonar, Multibeam Echosounders, Scanning Imaging sonar, Side Scan Sonar (SSS), Synthetic Aperture Sonar (SAS), Forward Looking Sonar (FLS), etc. [1][2][3][4][5][6][7]. However, SSS and SAS sonars are often considered more suitable for surveying vast areas owing to their long range and high resolution of sensing, whereas FLS are more suited for close-up and detailed inspection of the immediate local environment [1]. One approach described by Galceran Et al. consists of initially detecting possible objects in SSS/SAS imagery over a vast survey area and then performing re-acquisition of these objects utilizing FLS to further asses the detected objects.

Moreover, two types of FLS sonars are widely used, Multibeam and Sector Scanning. But, Multibeam FLS are more widely used in underwater environments for the kind of tasks that is being studied here [4]. Furthermore, recent developments of two-dimensional forward-looking sensors (2D-FLS) sonars have evolved, which provide high-definition acoustic images at near-video frame rates, which enable better association and tracking performance [5].

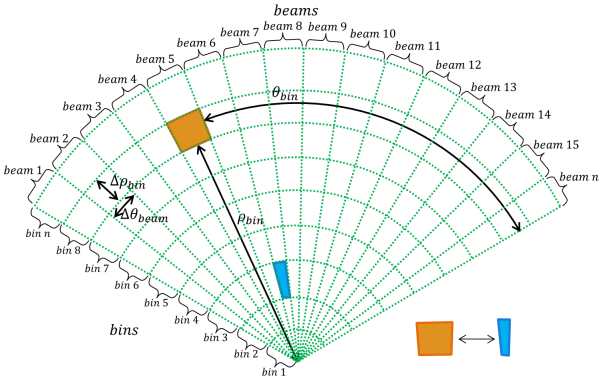
B. Imaging

The Sonar insonifies the environment with acoustic waves, spanning its field-of-view (FOV). The acoustic waves' returns are captured by the Sonar, specifically in the case of FLS using a horizontal array of transducers [2]. One way image information is generated is through a beamforming process, which orders the acoustic energy temporally (related to the distance of the echoes, thus providing the range) and the acoustic beam (providing the angle of arrival of the return) [2]. The imaging can hence be represented in Polar coordinates (range and bearing), which can be transformed into Cartesian coordinates (fan-shaped acoustic image) [2].

The acoustic waves that return from the same angle belong to the same beam, and are split into bins based on the range of the returns shown in Fig. 2b. Due to the specifications



(a) Sample sonar configuration, FOV breaking sea-surface [2]



(b) A representative image formation of an FLS. Each bin can be identified on the polar coordinate system (θ_{bin}, ρ_{bin}) and has an angular resolution $\Delta\theta_{beam}$ and a range resolution $\Delta\rho_{bin}$. [3]

Fig. 2: Sonar Configuration & Imaging

of FLS operation, the information pertaining to elevation is usually indistinguishable. Hence, the acoustic image can be considered as a 2D projection on the horizontal plane of the observed environment [3]. This, however, does not pose significant issues because the error introduced by the projection approximation holds if the elevation is negligible compared to the range [5].

Moreover, the Sonar's FOV is setup depending on the requirement of the imaging. For example, if detecting both surface and underwater objects are required, the FOV can be adjusted to cross the water surface as shown in Fig. 2a, whereas if only seabed objects are to be identified, a downward-looking angle would be more appropriate.

C. Challenges related to FLS Imaging

Although acoustic images produced by sonars are almost independent of water turbidity, noise and absorption of the waves can still distort imaging. Additionally, certain characteristics of the data increase the difficulty of handling and extracting information [3][5].

- *Low resolution*: Contrary to their optical counterparts, whose two-dimensional sensor arrays consists of millions of pixels, sonars images have a much lower resolution although still being high-definition [5].
- *Inhomogeneous resolution*: The number of pixels representing a bin decreases with range in the Cartesian space due to the polar nature of the sonar's sensors; visually described in Fig. 2b, the orange and blue patches represent the same resolution, but belong to different ranges and differ in physical areas. This causes an increase in measurement sparseness with range [3], and the non-uniform resolution causes distortion of the image in the cartesian space [5].
- *Low Signal-to-Noise Ratio*: Sonars usually suffer from low SNR due to speckle noise caused by mutual interference of the acoustic returns or any other source of noise naturally present in the environment [3][5].
- *Acoustic reverberation*: It is caused when multiple acoustic returns from the same object are captured, with the possibility of creating duplicates of the object in the image depending on the width of beams and spread of the returns [3].
- *Acoustic shadow effect*: The path of the acoustic waves can get blocked by obstacles, causing occlusion of objects and producing a black region devoid of any acoustic feedback [3]. It can also occur as a result of complete absorption of the waves by the medium or objects.
- *Inhomogeneous insonification*: Sonars can be affected by inhomogeneous intensity patterns caused by differing sensitivity of the transducers and the lenses, which can be alleviated to an extent by sampling a large number of images to estimate the patterns for the device in use [5].
- *Changes in viewpoint*: When the sonar generates images of the same region from different points of view, there can be enormous changes in the visual appearance of the images, which can cause issues related to the registration of images and tracking of objects [5].

IV. Object Detection

THE next step in the process is analyzing the sonar images to detect objects in the scene. Researchers explore the detection of objects from sonar images using traditional image processing techniques and some advanced deep learning techniques. Moreover, detection of objects relative to their location, i.e. surface or submerged, is explored since certain characteristics of the images differ between the two, hence requiring specialized solutions.

A. Detection of submerged objects using Integral Image

Galceran Et al. proposed a novel algorithm for the detection of underwater human-made objects in FLS imagery. It takes advantage of the integral-image representation, which provides fast and efficient computation of features. Furthermore, the computational load is reduced by working on small regions of the images enabling execution in real-time with limited computational resources [1]. The algorithm analyzes local regions to find echo-highlights that are higher than the local background. Apriori information about different objects, such as their shape and size, are used to filter. The steps involved in the algorithm, depicted visually in Fig. 3, are as follows:

- *Region of interest*: Rectangular regions of interest from the sonar image are extracted, which also helps avoid noisy and low-quality areas of the image. Every subsequent step in the algorithm is performed only on this extracted rectangular region.

- *Integral Image*: A representation that allows for fast computation of the sum of pixel values in a rectangular area of the image. For image A, the integral image I is:

$$I(x,y) = \sum_{x' \leq x, y' \leq y} A(x',y') \tag{1}$$

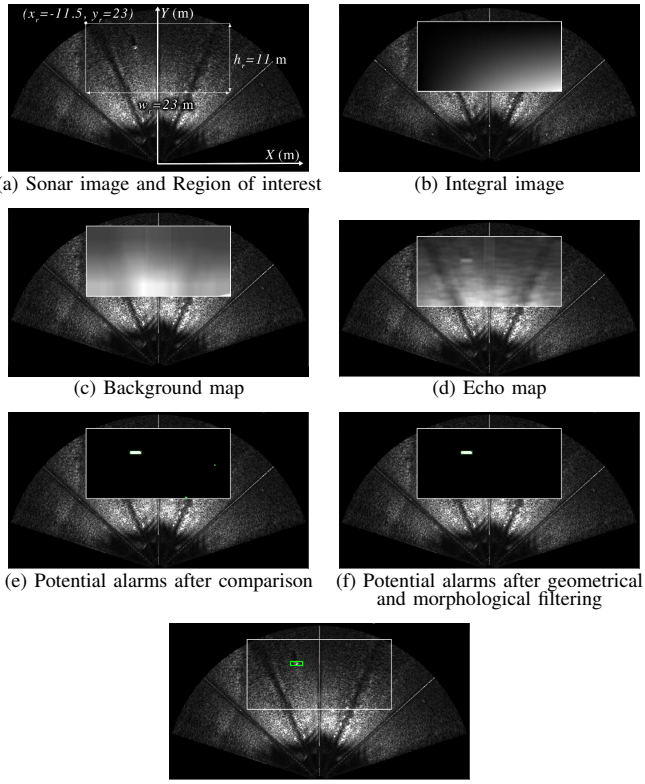
calculated in one pass over the entire image as:

$$I(x,y) = I(x-1,y) + z(x,y) \tag{2}$$

$$z(x,y) = z(x,y-1) + A(x,y) \tag{3}$$

- *Background Estimation*: The local Background Map is estimated using the integral image which establishes the seabed reverberation level [1]. Two different sized concentric sliding windows are used to calculate the mean pixel value of the neighboring pixels in the bigger window, ignoring the pixels in the smaller window [1].
- *Echo Estimation*: An Echo Map is constructed using the integral image to locate high intensity echo returns, where for each pixel the mean value of the neighboring pixels using a single sliding window is computed [1].
- *Potential Alarms Detection*: Any pixel [x,y], for which the echo map value, E[x,y] is sufficiently higher than the background map value B[x,y], i.e. $E(x,y) > \beta B(x,y)$ is declared to undergo further investigation, with β being the scaling factor adjusted as required [1].
- *Geometrical and Morphological Filtering*: The determined potential alarms are filtered according to their geometrical and morphological properties to rule out objects that do not correspond to objects that need to be detected [1].
- *Echo Scoring and Thresholding*: In the final step, the Echo Score S_i is calculated as the mean pixel value of all pixels in the blob for each remaining potential alarm. A detection threshold is determined according to preference to finalize detected objects [1].

Galceran Et al. empirically demonstrated the viability of the algorithm at the Autonomous Neutralization Trial (ANT'11). Multiple autonomous missions were executed in a circular trajectory around or cross pattern centered over known targets at varying depths (from 5m to 12m). Each detected target's location, calculated in the global coordinate frame, had errors in the range of 0.6m to 3.8m compared to their apriori information. A sample empirical result has been depicted in Fig. 3.



(a) Sonar image and Region of interest (b) Integral image

(c) Background map (d) Echo map

(e) Potential alarms after comparison (f) Potential alarms after geometrical and morphological filtering

(g) Submerged object detection algorithm using Integral image

Fig. 3: Submerged object detection algorithm using Integral image, Galceran Et al. [1]

B. Detection of Sea-surface objects

Karoui Et al. proposed a Sea-surface obstacle detection and tracking procedure for human-made objects, like buoys, boats, containers, etc. It is a hierarchical detection workflow, which carries out both detection and tracking together, and uses information extracted from previous images to improve detection in the current image under process [2]. In this section, we shall explore only the detection part of the workflow.

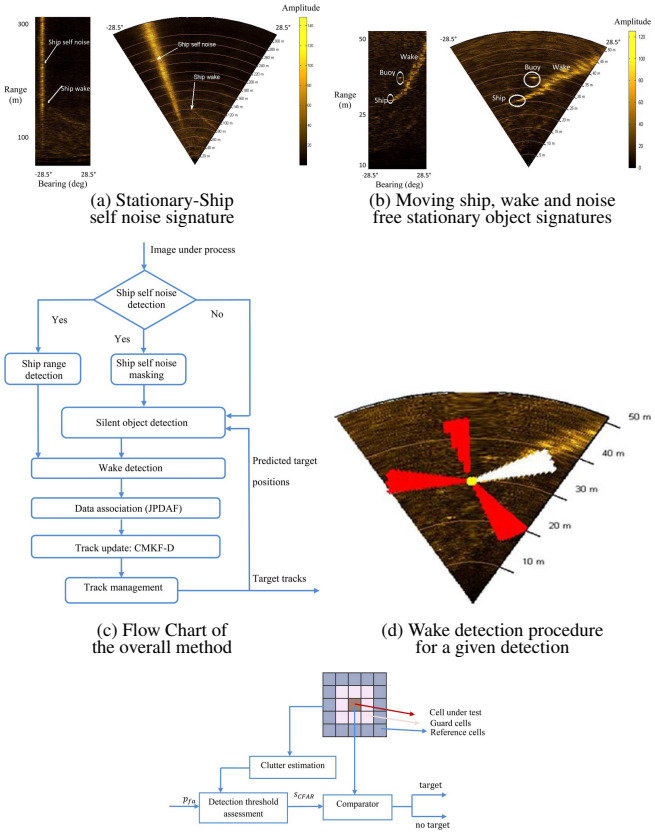
According to the type of object and it's state (still or moving), the surface acoustic signatures can be categorized as [2]:

- *Strong Intensity Beams* in the sonar image, due to stationary ship noise representing the ship bearing.
- *High contrast intensity features* for noise-free objects, such as buoys, sailboats, and static ships.
- *Some high-intensity lines due to wake* in the track of moving vehicles.

The proposed detection architecture is structured in three distinct steps as follows [2]:

1) Detection of Stationary Ships with Self-Noise: Since stationary ship noise is strong and could spread over several adjacent beams, it could cause issues with the detection of other features; hence it is detected first. Beams that are more energetic than their neighbors based on the evaluation of the median echo level along each beam are selected to provide the ship's bearing. To handle noise, a moving-window median filter is applied to these beams, and then the highest intensity pixels in these beams provide the ship range.

2) Detection of Static Noise-Free Targets: The Constant False-Alarm Rate (CFAR) detection algorithm with an adaptive threshold is used to calculate clutter estimate for each image cell, taking into account clutter-power estimates from surrounding Reference Cells, and ignoring the Guard Cells to ensure information from the same extended target under study is not used in the estimation, as described in Fig. 4e. Furthermore, to avoid noisy estimations, information is exchanged between the detection and tracking phases, and reference cells are ignored, which belong to already detected beams, wakes, and target positions.



(a) Stationary-Ship self noise signature

(b) Moving ship, wake and noise free stationary object signatures

(c) Flow Chart of the overall method

(d) Wake detection procedure for a given detection

(e) CFAR detector. p_fa is the probability of false alarm, and s_CFAR is the detection threshold

Fig. 4: Detection of Sea-surface objects, Karoui Et al. [2]

3) Wake Detection for Moving Ships: The CFAR algorithm discussed in the last step can detect wake cells in the image as well, whenever their amplitudes are high enough. Since the eventual goal is to identify the ship's actual position generating the wake, the wake ends are searched. The area around each CFAR detected cell is analyzed with oriented strips. The assumption is that if the amount of CFAR detections in one orientation is significantly higher than the other orientations around the cell, then it is likely to be a wake end. The computation time is reduced by carrying out wake-end detections only around cluster centers i.e., regions with closed spaced detections.

Karoui Et al. presented empirical results in four different sequences of sonar images. The analysis revealed false alarm rates of detection consistent with the theoretical possibility ($p_{fa} = 10^{-6}$) in the absence of wakes. In contrast, it significantly increased in the presence of wakes (related to small inflatable boats). Despite the limitations, there exists the advantage of reducing the computation time of the overall method, including tracking, as will be discussed further in Section VI. A sample empirical result for the three-step detection procedure is presented in the Fig.(s) 4a, 4b and 4d.

C. Detection of submerged objects using CNN

Valdenegro-Toro developed a Convolutional Neural Network (CNN) that reliably scores objectness of windows in FLS images, generating thresholded detection proposals, which works as a class-independent object detection technique. Unlike traditional Image Processing techniques where features are engineered, resulting in class-specific object detectors, CNNs learn representation directly from labeled data [7].

Given a labelled dataset containing objects of interest, a training set is constructed by running a sliding window over each image and cropping each window with an intersection-over-union (IoU) score above a predetermined threshold. Given two rectangles A and B, the IoU score is defined as:

$$IoU(A, B) = \frac{area(A \cap B)}{area(A \cup B)} \tag{4}$$

For each cropped window, the window's ground-truth objectness is estimated as a score based on the maximum IoU with ground truth. Capturing using sliding-windows generates multiple training windows that contain the same ground truth or a part of it, and the IoU score decreases as the window moves away from the ground truth. This essentially forms a data augmentation process, which diversifies the training data resulting in better generalization performance [7].

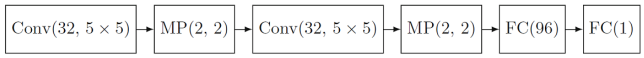
The CNN consists of 4 layers as described in Fig. 5a, which takes in as input 96×96 images. The first layer is a convolutional layer (Conv), with 32 5×5 filters, followed by a 2×2 Max-Pooling (MP). The second layer is identical to the first. Finally, the two classifier layers consists of one Fully Connected (FC) layer with 96 neurons and the final FC layer with one neuron. All layers use the ReLU activation function, except for the FC layers, that use a sigmoid function to output objectness score in the [0,1] range. Batch Normalization layers are inserted after every MP layer and between the two FC layers.

The network is trained using the Mean Squared Error (MSE) loss function, with a mini-batch gradient descent algorithm, using the ADAM optimizer. The batch size used was 32, with the initial learning rate $\alpha = 0.1$. The network is trained for 10 epochs, with an early stopping criterion based on validation loss. The result of testing the network to identify objects in sonar images is shown in Fig.(s) 5b and 5c.

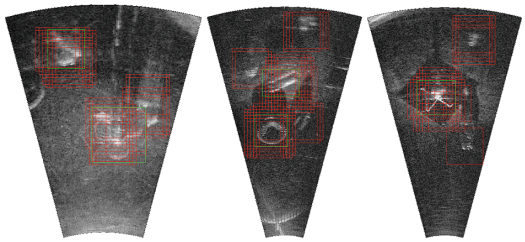
Valdenegro-Toro used an 85/15 split for dividing the training windows into training and validation sets and generated further testing datasets using the similar sliding window technique as before. The empirical results reported are promising, with 94% recall at a fixed threshold ($T_0 = 0.5$). Analysis of missed detections points to inaccurate localization caused by the sliding window for small objects. However, good generalization performance is reported by analyzing the testing results on samples not used in training.

V. Object Classification

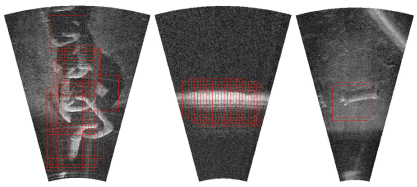
THE next task in the workflow is identifying these detected objects, primarily if class-independent detection proposals are generated. Real-world applications like generating a map of the environment with artifacts allow for skills like navigation, interaction with the environment, self-localization, etc. [3]. Hence, there exists an increasing interest in integrating such semantic knowledge with geometrical information [3]. In this study, we explore classification approaches using Machine Learning techniques applied to acoustic images.



(a) Neural Network Architecture



(b) Results on test images. Red rectangles are proposals, whereas green rectangles are ground truth.



(c) Results on out-of-sample images.

Fig. 5: Convolutional Neural Network based Objectness scoring and detection proposals, Valdenegro-Toro [7]

A. Segment Classification using Machine Learning techniques

Santos Et al. proposes an approach for classification comprising four significant steps described in Fig. 12a. The first three steps are techniques for enhancing, segmenting, and describing the acoustic images, required to alleviate the challenges associated with FLS as described in Section III-C.

1) Image Enhancement: The image correction is performed in this step. The inhomogeneous insonification of the images is mitigated by estimating the sonar's insonification pattern using a large pool of acoustic image samples from the sonar, and then correcting every image that needs to be used.

2) Image Segmentation: Since objects are more efficient at reflecting the acoustic waves than the seabed, they are characterized by high-intensity spots on the acoustic images. Hence, an approach based on sonar operation principles to detect peaks of intensity has been adopted. Then, the connected pixels are detected for each peak using a breadth-first search (BFS) algorithm, with an 8-way connection for the neighborhood criterion, which reduces issues related to multi-segmentation of a single object [3].

3) Describing segments: Segments are then described using a Gaussian probabilistic function. A covariance matrix, relating the x and y positions of each pixels of a segment, is generated. Singular Value Decomposition (SVD) is performed on the this matrix to compute the eigenvalues and eigenvectors, and the largest eigenvalue is defined as the $width$, and the second largest eigenvalue is defined as the $height$. Based on these, a 10-dimensional feature vector is defined for each segment, which consists of the following dimensions:

- *Height*
- *Width*
- *Inertia Ratio*, i.e. (width / height)
- *Mean* of the acoustic returns
- *Standard Deviation* of the acoustic returns
- *Segment Area* calculated using Green's Theorem
- *Convex Hull Area*
- *Convexity*, i.e. (segmented area / convex hull area)
- *Perimeter*
- *Number of pixels*

4) Segment Classification: Once the 10-D features are generated for each segment, they are classified using supervised classification algorithms. Santos Et al. evaluated three different Machine Learning techniques, namely Support Vector Machines (SVM), Random Trees (RT), and K-Nearest Neighbors (KNN), using five different classes of objects (pole, boat hull, stone, fish, and swimmer).

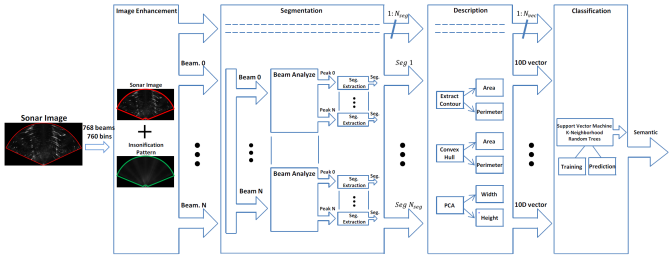
- *Support Vector Machine*: SVM classifier computes the optimal hyperplane, using optimization techniques with appropriate loss function, that best separates the n-dimensional training vectors into their corresponding class. Santos Et. al uses grid search techniques to determine the best non-linear kernel parameters (γ, C).
- *Random Trees*: RT is an ensemble learning method that combines multiple learning models. In RTs, a collection of random Decision Trees (DTs) are generated to predict the object's class, based on a majority vote approach. Each DT in the collection is trained using the same set of parameters but on different sample datasets generated randomly from the original data population.
- *K-Nearest Neighbors*: KNN is a simple non-parametric algorithm with no training phase, and uses appropriate distance metrics for new and unknown feature vector to be predicted. A new vector's class is predicted using a majority vote approach of the K nearest neighbor vectors' classes. The value of K is the only control feature to check for the generalization of the method.

The empirical results presented by the researchers have been generated offline, using the ARACATI 2014 dataset [3]. Alongside the 10D feature set, all possible 2D combinations of the 10 dimensions mentioned before were also used to benchmark the ML models. To mitigate the issues of an unbalanced dataset, bootstrapping techniques were adopted to create a balanced dataset as well, and the experiment was performed on both datasets. The 2D feature vectors could only reach a maximum hit-rate of about 89.83% using Random Trees. However, the 10D feature vectors achieved marginally better results with a 93.57% hit-rate with the KNN classifier with $K = 1$, followed by SVM with RBF kernel at 89.90%, and Random Trees at 78.89%.

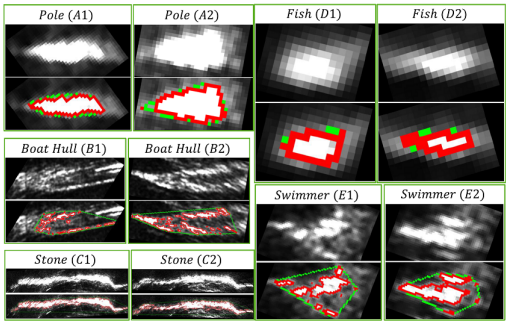
A sample of the empirical results has been presented in Fig. 6. Segmentation of various objects like Pole, Boat hull, Stone, Fish and Swimmer, and the classification results of one ping have been shown in Fig. 12b and Fig. 12c respectively.

VI. Object association and tracking

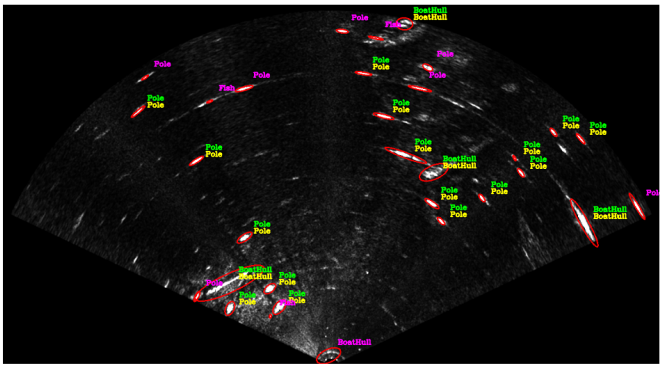
In a dynamic environment, detected objects need to be tracked efficiently in real-time, enabling quick interpretation of the environment by the AUV. Tracking objects enables detecting potential obstacles in the AUV's path, leading to safe



(a) Overview of the proposed semantic system.



(b) Segmentation results in Cartesian coordinates. Pixels in red represent the segment contour, and green represent the convex hull.



(c) Sample acoustic image, annotated. Yellow labels are manually defined; green labels indicate correct classification, red indicates incorrect, magenta indicates segments without annotation to compare.

Fig. 6: Image Segment Classification with ML, Santos Et al. [3]

and effective traversal capabilities. However, before tracking can be performed effectively, there is a need for the images or objects to undergo registration or association over the pings.

A. CMKF-D and JPDAF based association and tracking

Karoui Et al. demonstrates the use of Converted Measurement Kalman Filter with Debiased conversion (CMKF-D) for tracking the objects in the scene, with the use of Joint Probabilistic Data Association Filter (JPDAF) for the association of objects to their correct tracks, the overview of which is detailed in Fig. 4.

1) Object Association: Karoui Et al. proposes the use of JPDA Filter due to its comparatively lower computational costs, robustness in noisy environments, and suitability in tracking multiple targets. In this approach, the innovation vectors ϑ_k^t, used to update the state, are estimated by combining every measurement which lies in the target validation gate. The gate represents an ellipsoid region for every target, which is computed at every sampling time to select measurements that have a higher probability of being correct.

2) Object tracking: The CMKF-D consists of the two standard Kalman Filter steps, the prediction and the update steps, which are iterated over for each target t. The polar coordinates are converted to Cartesian coordinates to work with a linear measurement equation (6). The bias that arises from the non-linearity of measurement conversion is handled by the CMKF-D. The tracking of targets is performed in the Cartesian frame according to the classic near-constant velocity dynamical model:

$$x_k^t = F x_{k-1}^t + G v_{k-1} \tag{5}$$

where, x_k^t are the position and velocity vectors in the xy plane relative to the sonar, v_{k-1} is a 2-D zero-mean independent and identically distributed (IID) white Gaussian noise [2]. Matrices F and G of the state-equation (5) encode time-sampling. At ping k, measurements that originated from target t are related to the state vector x_k^t according to the following measurement equation:

$$y_k = H x_k^t + w_k \tag{6}$$

where, H is a known matrix and w_k is a known measurement noise [2]. Moreover, since JPDA Filters do not allow automatic track initiation and termination, a rule-based heuristic approach was adopted by Karoui Et al. to manage the tracks of targets.

The empirical results presented by the authors are only a qualitative assessment since GPS enabled ground truth was unavailable. In the four different sequences of recorded data that the researchers experimented upon, the proposed technique successfully tracks various objects like concrete blocks, buoys, sailboats, etc. However, false tracks did occur due to artifacts at the edge of the sonar image. Tracking results of different detected objects have been depicted in Fig. 7.

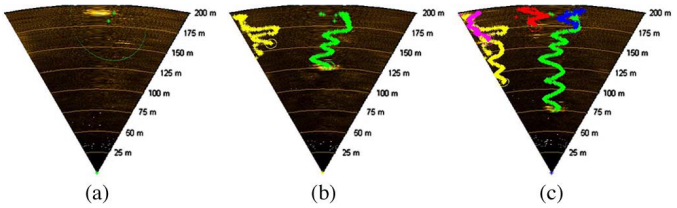
B. Occupancy Grid based association and tracking

Ganesan Et al. also proposes a Bayesian Filtering-based tracking approach but uses Local Occupancy Grids to represent the belief of the location of nearby objects and require a motion and measurement model to update the occupancy grid accordingly.

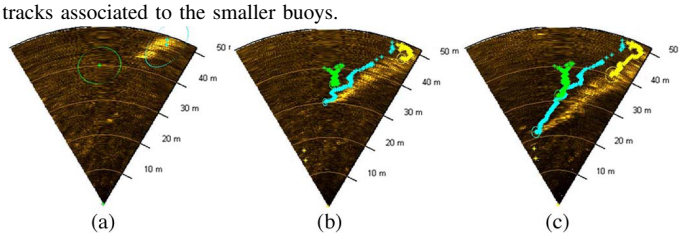
1) Occupancy Grid: To deal with noisy data, Occupancy grids are considered to be better suited since they associate a probability of occupancy to every cell on the grid, instead of using thresholded intensity values to indicate a detection [4]. There are two major types of occupancy grids widely used:

- *Global Occupancy Grids* are used to create a comprehensive map of all the detected features and obstacles in a global frame of reference while accounting for increasing positional error [4].
- *Local Occupancy Grids* are attached to the AUV's frame of reference. When the AUV travels, objects in its vicinity move in the grid relative to the AUV's motion [4].

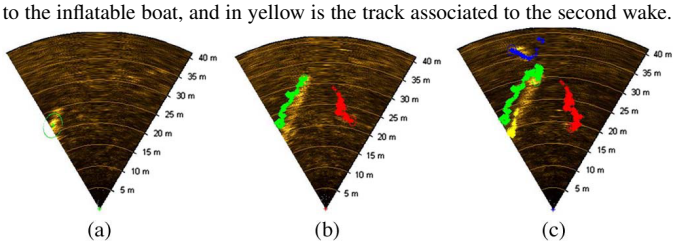
Ganesan Et al. propose that Local Occupancy Grids are sufficient for the studied use case since it localizes objects accurately with respect to the AUV.



(a) In green is the track associated to the large buoy, yellow and magenta tracks correspond to the two concrete blocks, and in red and in blue are the tracks associated to the smaller buoys.



(b) In green is the track associated to the buoy, in cyan is the track associated to the inflatable boat, and in yellow is the track associated to the second wake.



(c) In green is the track associated to the inflatable boat, in yellow is the track associated to the second wake endpoint, in red is the track associated to the fixed buoy, and in blue is the track related to a sailboat.

Fig. 7: Tracking result over various pings, Karoui Et al. [2]

The Local Occupancy Grid has been defined as a rectangle of size $m \times n$, divided into occupancy cells of size $l \times l$, with its location fixed with respect to the AUV, as illustrated in Fig. 8a, and represents the *belief* that the algorithm holds. It is represented by a matrix P, which contains the probability that the occupancy cell is occupied, denoted by $[P[O(x, y)] \forall x, y]$.

Each of the acoustic returns from the environment is discretized into a set of bins (k, θ) (representation similar to Fig. 2b), where k is the range and θ is the bearing, with the observation for each bin being $z_{k,\theta}$. With a threshold value t_k for a range bin k, the detection $S_{k,\theta}$ is reported as 0/1. The probability of detecting an obstacle for a range bin k as p_k and the probability of false alarm as f_k have been defined, along with a constant acceptable false alarm rate f.

2) Motion model: The occupancy cells' probabilities are updated using a motion model that accounts for both translational and rotational motion of the AUV. The authors propose a decoupling of the two motions, which allows for the real-time performance of the algorithm [4].

- *Translational Motion* of the AUV has been modeled as an application of convolution operation on the cell probabilities using an appropriate kernel K, which is chosen based on whether the AUV's motion is deterministic or probabilistic in nature [4]. It is represented as $P_t = P_{t-1} \otimes K$, where $\otimes$ represents convolution.
  - *Deterministic Motion* is applied when GPS or DVL signals are available, which are highly accurate with low noise. The occupancy grid is shifted by the amount of displacement using a chosen kernel based on the amount of displacement undergone [4].
  - *Probabilistic Motion* is applied otherwise, where the kernel represents displacement as a uni-modal Gaussian distribution, with the mean translational motion being denoted by the peak, and the spread denoting the uncertainty associated with the motion estimates [4].
- *Rotational Motion* of the AUV has been modeled as deterministic due to the high accuracy of compasses. Changes of the heading are accumulated until they reach $\pm 1^\circ$ to avoid errors related to rounding off [4].

3) Measurement model: The occupancy grid P serves as the Bayesian prior, updated using a measurement model when a new measurement becomes available. Bayes' rule and the different probabilities (p_k, f) have been used for updating the values of P to their posterior. Heuristics like the region of overlap between occupancy cell (x, y) and range bin (k, θ) denoted by $O^{x,y}_{k,\theta}$, and it's area $v^{x,y}_{k,\theta}$ have been further adopted.

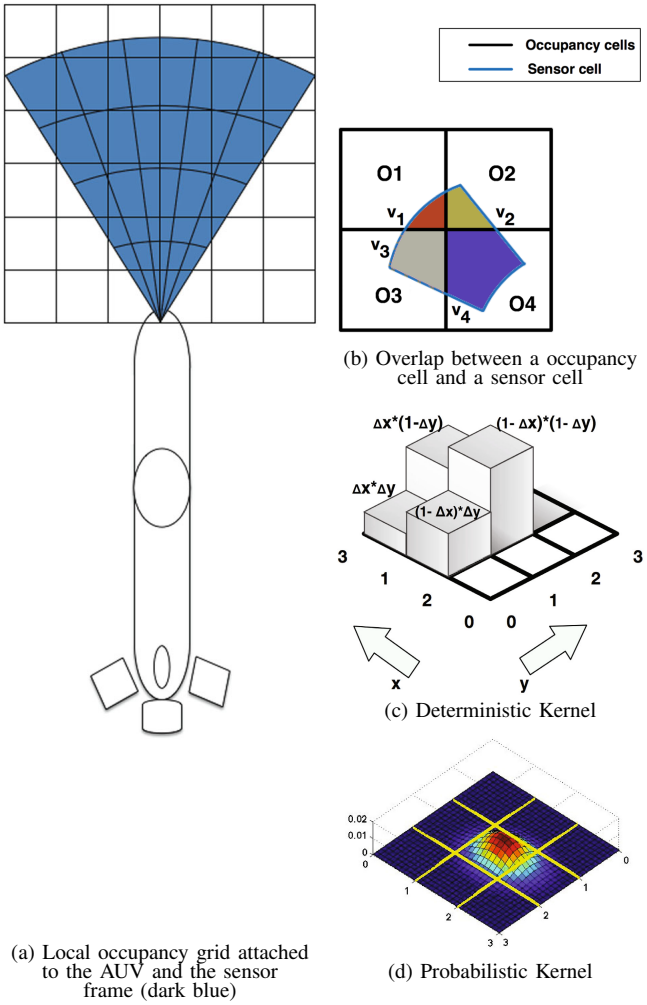
Ganesan Et al. conducted experiments at two different locations. The results suggest that noise followed a Gaussian distribution at the reservoir, whereas a right-skewed Stable distribution at sea. Moreover, the probabilities and false alarm rates (p_k, f) required adaptation based on the environmental conditions. Furthermore, information from multiple scans was processed to improve target separation from background noise. Moreover, an implicit assumption of events where occupancy cells are occupied independent of each other i.e. $P(O_{i,j}|O_{x,y}) = P(O_{i,j})$ has been made. This, however, might not be universally true, for example, in the case where the size of an object is larger than the occupancy cell.

C. Adaptive Particle Swarm Optimization

Although Particle Filters (PF) are widely used for tracking purposes and are robust owing to the adaptive variance of noise decreasing the influence of the environment, they suffer from immense computational costs due to the large number of particles required [6]. Similarly, Kalman Filter based approaches seem to have some limitations in underwater scenarios, especially when non-linear motion dynamics and observation processes are involved [6]. Wang Et al. propose an Adaptive Particle Swarm Optimization (APSO) algorithm to track underwater targets as follows:

1) Adaptive Inertia Weight: Due to the problem of diversity loss and premature convergence in the generic Particle Swarm Optimization (PSO) algorithms, an adaptive inertia weight w has been proposed, where large w prevents the particles from falling into local optimum and a small w enables convergence to the optimal solution. Moreover, for each particle in the swarm, small fitness value defines strong exploration ability to converge on a global optimal solution, whereas large fitness value defines strong exploitation ability to keep particles close to the global optimal solution [6]. Hence, the adaptive inertia weight is defined as:

$$w = (1 + cos(\frac{t \cdot \pi}{t_{max}})) \cdot (1 - \frac{f}{f_{max}}) \cdot w_{ini}/2 + w_{min} \quad (7)$$



(a) Local occupancy grid attached to the AUV and the sensor frame (dark blue)

(b) Overlap between a occupancy cell and a sensor cell

(c) Deterministic Kernel

(d) Probabilistic Kernel

Fig. 8: Occupancy Grid based detection artifacts, Ganesan Et al. [4]

where, t is the current iterative times, t_{max} is the maximum number of iterations, f is the fitness value of the current particle, f_{max} is the global optimal fitness value, w_{ini} is the initial inertia weight, w_{min} is the minimum inertia weight [6].

2) Velocity update of particles: To further solve the issue of particles falling into local optimum, updating the velocity of the particle based not only on individual and global optimal information but also using a random particle selected from the swarm has been proposed. The velocity and position update of particles is formulated as:

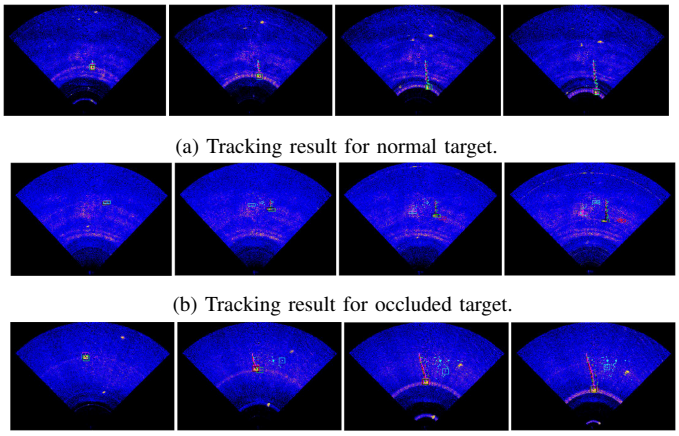
$$\begin{aligned} v_{ij}^{t} = w \cdot v_{ij}^{t-1} &+ c_1 \cdot r_1 \cdot (pbest_{ij}^{t-1} - x_{ij}^{t-1}) \\ &+ c_2 \cdot r_2 \cdot (gbest_{ij}^{t-1} - x_{ij}^{t-1}) \\ &+ c_3 \cdot r_3 \cdot (rbest_{ij}^{t-1} - x_{ij}^{t-1}) \end{aligned} \tag{8}$$

$$x_{ij}^{t} = x_{ij}^{t-1} + v_{ij}^{t} \tag{9}$$

where, c_1, c_2 and c_3 are the learning factors defined as:

$$c_i = 2.8 \cdot \frac{fit_i}{fit_1 + fit_2 + fit_3} \tag{10}$$

where, fit_1 is the optimal fitness value of the current particle, fit_2 is the global optimal fitness value, and fit_3 is the optimal fitness value of a random particle r [6].



(a) Tracking result for normal target.

(b) Tracking result for occluded target.

(c) Tracking result for target affected by noise.

Fig. 9: Adaptive Particle Swarm Optimization tracking results, Wang Et al. [6]

3) New Update Strategy: To handle occlusion, the proposed technique regenerates new particles according to the level of occlusion, to explore the target position and quickly relocate, and guarantee diversity of particles in the swarm. The new update strategy is defined as:

$$x_{ij}^{t} = \begin{cases} x_{ij}^{t-1} + v_{ij}^{t}, & 0 < R < H(f) \\ X, & H(f) \leq R < 1 \end{cases} \tag{11}$$

where, $H(f)$ is a parameter that is adjusted according to the level of occlusion with its value in the range [0,1], R is a random number in the range [0,1], X represents the random position in the space. The value of H_f has been adapted to control the probability of regenerated particles in order to solve the occlusion problem, by dividing the fitness values into three different regions. The parameter $H(f)$ is given by:

$$H(f) = \begin{cases} 0, & f^{t-1} < F_{min2} \\ H(f) \cdot f^{t-1}, & F_{min2} \leq f^{t-1} < F_{min1} \\ H(f), & else \end{cases} \tag{12}$$

where, f^{t-1} is the fitness value of the current particle in the last frame; F_{min1} and F_{min2} are the predefined thresholds to identify occlusion, $F_{min2} < F_{min1}$ [6].

4) Underwater target tracking using APSO: For application of the proposed APSO algorithm, the target for tracking is selected using a rectangular region. The features of the target and particle region are obtained from their respective information. In each iteration, feature extraction is done per particle using the Hu moment invariant feature, and the similarity with the target feature is used as the fitness value of the particle, and then the target is tracked using Eq. (11) and Eq. (12). The fitness function is defined as a correlation coefficient, expressed as:

$$f = \frac{\sum_{k=1}^{n} H_1(k) H_2(k)}{\sqrt{\sum_{k=1}^{n} H_1(k)^2} \sqrt{\sum_{k=1}^{n} H_2(k)^2}} \tag{13}$$

Wang Et al. empirically gathered evidence of the tracking results for various different types of objects. In Fig. 12, the results associated with tracking an usual target, an occluded target and a noisy target has been shown.

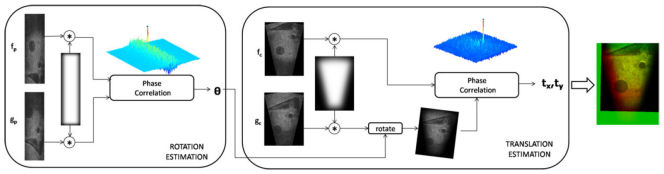


Fig. 10: Overall registration pipeline, Hurtós Et al. [6]

D. Fourier-based Registration

Image registration techniques are alignment methods to transform different sets of data into one global coordinate system. Image registration techniques are applied to navigation and mapping tasks, where it is a crucial step in the mosaicing, and motion estimation applications [5]. Hurtós Et al. propose a Fourier-based image registration technique on 2D FLS imagery, a global method that considers the whole image content contrary to feature-based approaches in most image processing techniques and contributes to the minimization of ambiguities in the registration process.

In the proposed methodology, the registration technique computes pairwise constraints between pings, which are then embedded into a pose-based graph estimation for global alignment. This enables the rendering of consistent 2D acoustic mosaics of high detail that offer a global overview of the AUV's survey area while providing significant improvements to the SNR and resolution with respect to the individual acoustic images. Furthermore, the registration technique is used to extract 2D AUV motion estimates.

1) Pairwise registration of FLS images: Hurtós Et al. proposes using a particular Fourier-based method, called the phase correlation algorithm since it allows registration with high computational efficiency due to the use of Fast Fourier Transform (FFT). As modeling in the frequency domain offers a much higher sensitivity to noise, a small smoothing filter is applied in the spatial domain to reduce noise and enhance the robustness of detected peaks [5]. The Fourier shift property, where a shift between two functions (defined as images here), is transformed in the Fourier domain into a linear phase shift as follows:

$$f(x, y) = g(x - t_x, y - t_y) \tag{14}$$

where, f(x,y) and g(x,y) are two images related by a two dimensional shift (t_x, t_y) [5]. The 2D Fourier transforms of the images are related as follows:

$$F(u, v) = G(u, v)e^{-i(ut_x + vt_y)} \tag{15}$$

where, F(u,v) and G(u,v) are the Fourier transforms of f(x,y) and g(x,y) respectively.

As described in Fig. 10, the pipeline outlines the procedure for registration of two images. Translational motion is recovered directly as a linear shift in the Fourier domain. However, rotational motion is mapped as a shift displacement directly on the polar images [5]. For loop-closure situations, where only rotational motion exists, a measure quantifying the registration's uncertainty is used [5].

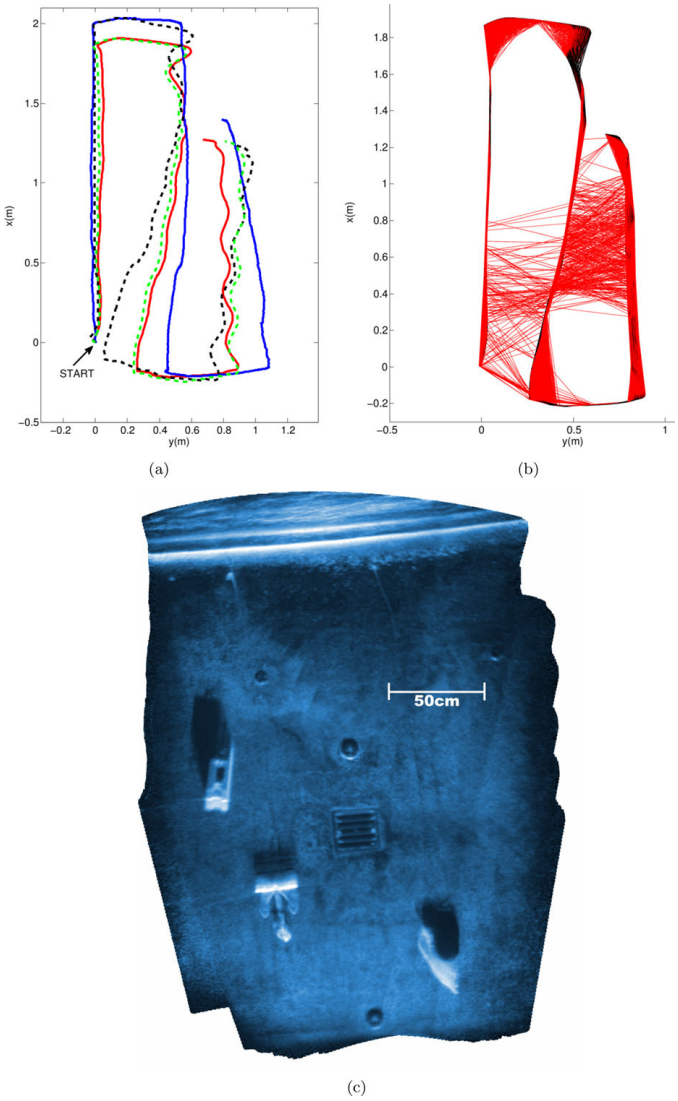


Fig. 11: The ARIS tank experiment results. (a) Trajectories. Blue: Vehicle's dead-reckoning trajectory. Black-dashed: Trajectory estimation from consecutive image registrations. Green-dashed: Trajectory estimation from the consecutive constraints including a window of local neighbors. Red: Final estimated trajectory after the global alignment. (b) Final graph constraints. Black: window constraints. Red: loop-closure constraints. (c) Mosaic composition with 527 frames. Hurtós Et al. [6]

2) Global alignment: The generation of a global map of the environment is reshaped as a pose-based graph optimization problem, where a least-squares minimization is formulated to estimate the maximum-likelihood configuration of the sonar images, based on the pairwise constraints between different registrations, and a heuristic to quantify the degree of confidence in the alignment [5]. Since this can be computationally expensive, registrations are attempted only between frame pairs that are likely to overlap and are detected by first inferring the path topology based on consecutive image frames, and a wider window of neighboring frames [5].

Hurtós Et al. provided experimental results of the proposed methodology verifying the method's viability in three different settings, demonstrating results close to a globally-aligned tra-

jectory. The rendered mosaics are highly consistent, detailed, and accurate, enabling the identification of various objects and the environment. One experimental results in the controlled tank setting with the ARIS FLS has been presented in Fig. 11.

VII. Obstacle Detection and Avoidance

AFTER objects have been tracked successfully, their direction of motion and other measurements are incorporated to determine if an object becomes a threat of collision to the AUV. In such a case, avoidance maneuvers are executed for safe traversal of the AUV.

A. Detection & Local avoidance approach

As proposed by Ganesan Et al., the local occupancy grid is used to detect obstacles and send information to a command and control (C2) system of the AUV to execute avoidance maneuvers if needed. For detection of the obstacles, a threshold and a detection neighborhood is defined; and the rationale presented is that obstacles might not be confined to a particular occupancy grid and has the possibility of having moved since the detection procedures are only executed at the end of every scan.

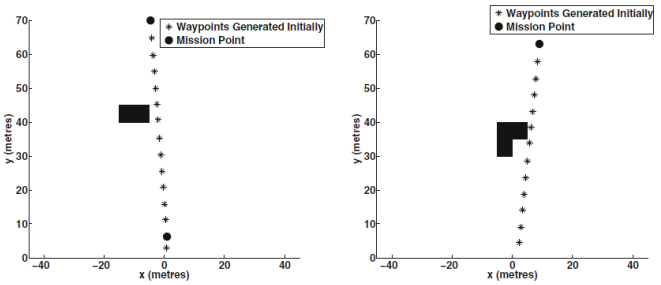
The C2 architecture is based on a hybrid hierarchical control architecture, which adopts an agent-based deliberate-reactive system [4]. The agents are as follows:

- *Captain* is responsible for starting and coordinating missions.
- *Executive Officer* receives mission points from the Captain and sends them to the Navigator for planning waypoints.
- *Navigator* plans waypoints, in the global frame, to a mission point and sends them to the Executive Officer, which forwards it to the Pilot.
- *Pilot* receives the waypoints from the Executive Officer and executes them systematically by defining vehicle parameters such as bearing, speed, depth, and altitude.

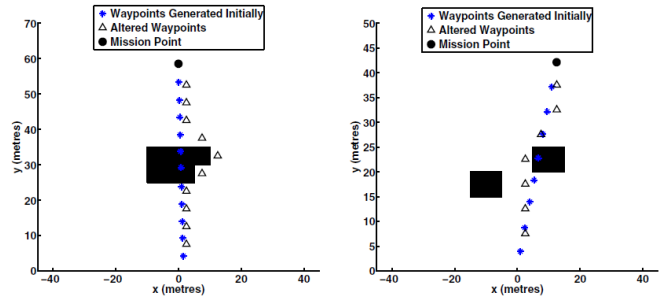
Ganesan Et. al proposes the addition of a new agent called *FLS detector*, and the modification of the *Navigator* agent to be well adapted for the task at hand as follows:

1) FLS detector: This agent directly communicates with the FLS and receives scan lines continuously from the sonar, and processes it to generate the local occupancy grid as described in Section VI-B. A suitable detection procedure is applied to detect obstacles in the vicinity of the AUV [4]. Thus, with the AUV as the reference frame, at the end of every scan, a detection map is generated which is sent to the Navigator [4].

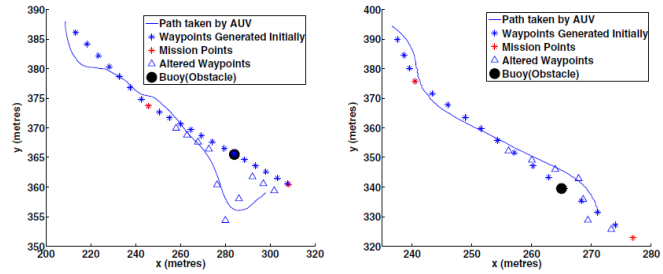
2) Navigator: Once the Navigator receives a detection map of the environment, it creates a new map, called the obstacle map. It provides a clearance radius to the obstacle, defined such that the cells in the clearance radius are a no-go zone for the AUV [4]. Since the obstacle maps are in the local frame, they are transformed into the AUV's frame of reference and then analyzed for possible collisions between the waypoints and the obstacles in the detection map, while also eliminating uncertainties associated with the position of the AUV [4]. Collisions are confirmed if waypoints or the line connecting



(a) Collision tracking by the navigator.



(b) Waypoint re-planning by the Navigator.



(c) Experimental results of obstacle avoidance.

Fig. 12: Occupancy Grid based Obstacle Detection and Avoidance results, Ganesan Et al. [4]

two successive waypoints lie on the obstacle or the no-go zones [4].

After detecting collisions, the Navigator immediately replans to create a new set of waypoints to the next mission point using the *A* Search* algorithm [4]. The new waypoints are then transformed back to the global frame of reference to be executed by the Pilot. The planning of the waypoints in the AUV's frame of reference makes the newly generated waypoints insensitive to the positional error growth of the AUV, hence even with uncertainty regarding the global alignment of the AUV can still safely execute an avoidance maneuver [4].

VIII. Conclusion

IN this study, various proposed techniques, and their empirical evidence has been explored related to the acoustic image-based object detection, classification, registration, and tracking of the AUV in an underwater environment. Furthermore, there seems to exist a pipelining capability of various related tasks applied to sonar imagery described in Section II.

Although the techniques studied here address some issues of previous methods, caveats still exist to these various approaches, and possible future development prospects for alle-

viating different concerns have been proposed by the authors. One concern is that the proposed techniques do not showcase any pre-processing of the images to explicitly remove noise instead of handling noise in the techniques themselves, which could provide more stable results; hence needs further empirical study. Similarly, the application of advanced techniques from Machine Learning and Deep Learning can be further studied for the subject's future exploration, such as deep learning techniques for classification, online object tracking, and other tasks.

Furthermore, sensor fusion techniques involving sensors for roll, yaw, and pitch can correct the deviation in trajectories that appear in the results of experiments, thus improving accuracy for safer traversal of the AUV [2]. Moreover, similar sensor fusion techniques could provide further heuristics for improving the already proposed techniques, which has not been brought under this study's umbrella.

REFERENCES

[1] Galceran, Enric. (2012). A real-time underwater object detection algorithm for multi-beam forward looking sonar. 306-311. 10.3182/20120410-3-PT-4028.00051.

[2] Karoui, Imen Quidu, Isabelle Legris, Michel. (2015). Automatic Sea-Surface Obstacle Detection and Tracking in Forward-Looking Sonar Image Sequences. IEEE Transactions on Geoscience and Remote Sensing. 53. 1-10. 10.1109/TGRS.2015.2405672.

[3] dos Santos, Matheus Ribeiro, Pedro Otávio Núñez, Pedro Drews-Jr, Paulo Botelho, Silvia. (2017). Object Classification in Semi Structured Enviroment Using Forward-Looking Sonar. Sensors. 17. 2235. 10.3390/s17102235.

[4] Ganesan, Varadarajan Chitre, Mandar Brekke, Edmund. (2016). Robust Underwater Obstacle Detection for Collision Avoidance. Autonomous Robots. 39. 10.1007/s10514-015-9532-2.

[5] Hurtós, Natàlia Romagós, David Cufi, Xavier Petillot, Yvan Salvi, Joaquim. (2015). Fourier-based Registration for Robust Forward-looking Sonar Mosaicing in Low-visibility Underwater Environments. Journal of Field Robotics. 32. 10.1002/rob.21516.

[6] Wang, Xingmei Wang, Guoqiang Wu, Yanxia. (2018). An Adaptive Particle Swarm Optimization for Underwater Target Tracking in Forward Looking Sonar Image Sequences. IEEE Access. PP. 1-1. 10.1109/ACCESS.2018.2866381.

[7] Valdenegro, Matias. (2016). Objectness Scoring and Detection Proposals in Forward-Looking Sonar Images with Convolutional Neural Networks. 9896. 209-219. 10.1007/978-3-319-46182-3_18.