DELFT UNIVERSITY OF TECHNOLOGY

RO MSC THESIS
RO57035

Multi Object Tracking for Unmanned Surface Vessels using LiDAR-AIS Fusion

Author:
Yohan J.P. Le Gars

Supervisors:
Dr. H. Caesar
Ir. L. van Litsenburg

November 28, 2024

Multi Object Tracking for Unmanned Surface Vessels using LiDAR-AIS Fusion

Yohan J.P. Le Gars
Supervised by Holger Caesar and Luuk van Litsenburg

***Abstract*—Autonomous navigation for Unmanned Surface Vessels (USVs) is essential for expanding their applications in maritime environments, where situational awareness and collision avoidance are critical. However, these environments present unique challenges for multi-object tracking (MOT) such as a wide diversity of vessel sizes, a high dynamic environment due to waves and current, and low visibility. Although neural network-based models could facilitate MOT in such scenarios, there is a lack of publicly available datasets in Maritime environments. Therefore, this work presents a probabilistic, point-based MOT framework specifically designed for short-range tracking ($\leq$ 100 m), utilizing weather-resilient sensors to ensure robust operation in diverse conditions. The framework integrates LiDAR and Automatic Identification System (AIS) data through a late fusion approach, improving state estimation by combining the dynamic tracking abilities of LiDAR with AIS's vessel identification capabilities. Key methods include an Interacting Multiple Model (IMM) for adaptive maneuver handling and Joint Probabilistic Data Association (JPDA) for data association. Validation in a simulated environment highlights significant limitations, showing that while the framework can manage basic tracking tasks, it remains far from optimal for the full scope of nearshore and coastal applications. This thesis underscores the need for further research to meet the demands of maritime MOT, particularly in handling the unique challenges posed by large vessels.**

keywords – MOT, USV, LiDAR, AIS, JPDA

I. Introduction

The maritime industry is undergoing significant advancements, with a growing interest in autonomous applications. In line with this progression, Unmanned Surface Vessels (USVs) have emerged as a key technology in autonomous maritime applications, supporting various payload configurations to adapt to the diverse requirements of marine surveying (bathymetric, hydrographic), which demand specialized equipment for different environments, ranging from shallow coastal waterways to the deep ocean [1]. Additionally, USVs offer significant operational advantages over traditional methods by improving safety, reducing costs, and enabling continuous, unmanned operations for the structural inspection and maintenance of maritime infrastructure [2]. This includes essential tasks at wind farms, offshore oil and gas facilities, bridges, and ports, where USVs can operate in hazardous or hard-to-reach areas, potentially working in conjunction with Remotely Operated Vehicles (ROVs) and Unmanned Aerial Vehicles (UAVs). To maximize these benefits, USVs must navigate autonomously while adhering to the International Regulations for Preventing Collisions at Sea (COLREGS) [3]. A key aspect of collision avoidance is maintaining situational awareness, which involves accurately mapping the static environment and tracking nearby moving obstacles with the help of sensors, a process known as Multi-Object Tracking (MOT) [4].

However, the maritime domain poses unique challenges to MOT due to its harsh environment (wind, waves, rain), impacting the perception capabilities of autonomous vessels. For instance, camera systems are particularly vulnerable in such conditions, despite advancements in computer vision, making them less reliable for maritime applications. As such, environmental perception at sea relies heavily on weather-resistant sensors like marine radars and the Automatic Identification System (AIS). X-band radar, known for its long-range detection capabilities, is commonly equipped on ships due to its effectiveness in spotting obstacles even in adverse conditions like rain and fog. AIS-equipped vessels broadcast their position with GPS-level accuracy, along with identity, motion information, and size. This heterogeneous sensor network is frequently employed by large ships to create a comprehensive maritime picture for enhanced situational awareness and semi-autonomous navigation [5]. However, radar performance declines in close-range detection (<250m), particularly for smaller objects such as buoys, navigation marks, and small boats, making it incompetent for MOT in nearshore environments [6]. Additionally, AIS has its own limitations due to its low and inconsistent update rate, ranging between 1 and 20 seconds. AIS signals can also be subject to interference from bandwidth sharing among vessels, leading to conflicting or missing AIS identifiers [7]. Furthermore, not all vessels are equipped with AIS, further reducing its reliability as a sole tracking solution [8]. In recent years, LiDAR (Light Detection and Ranging) technology has gained significant interest in maritime surveillance due to its precision, high data density, and its ability to operate independently of lighting conditions. In open waters, most LiDAR signals are absorbed by the water surface, resulting in point clouds that primarily capture above-surface obstacles, which is advantageous as it eliminates the need for ground removal. Additionally, the detection range of commercial LiDAR systems effectively addresses the blind spots left by marine radars, offering more reliable detection and tracking of nearby objects. In parallel, trained neural networks have become a popular tool for object detection and tracking in LiDAR point clouds, particularly in the automotive industry, where they have achieved consistent success. However, similar methods have yet to be widely applied in maritime environments, and publicly available labeled LiDAR

Delft University of Technology, Faculty of Mechanical Engineering, Department of Cognitive Robotics, Mekelweg 5, 2628 CD, Delft, The Netherlands.

point cloud datasets specific to maritime applications remain unavailable for training deep learning models. As a result, existing approaches in this field often rely on traditional probabilistic frameworks. Segmenting 3D LiDAR point clouds into objects of interest in such environments poses significant challenges due to factors like noise, sparsity, and varying point densities caused by the ego vessel's motion. Additionally, the irregular shapes of objects, sharp features, and absence of a consistent statistical distribution further complicate segmentation efforts. In maritime scenarios, the diverse range of vessel sizes exacerbates issues such as over-segmentation and under-segmentation, making accurate detection and tracking even more challenging [9].

This work addresses key challenges by utilizing AIS data to enhance LiDAR-based Multi-Object Tracking (MOT), aiming to mitigate issues such as self-occlusion, over-segmentation, and dynamic segmentation. It also contributes to the development of a framework designed for real-time deployment on an Unmanned Surface Vehicle (USV). A LiDAR-AIS MOT framework has thus been created with real-time processing in mind. Additionally, a simulated dataset, featuring various target vessels and maneuvers, has been developed to assess the performance of the MOT framework. Lastly, a ROS 2 node has been implemented to format both the results and ground truth, enabling the use of the nuScenes SDK for evaluating the framework with standard MOT metrics.

II. Related Work

In the context of Multi-Object Tracking (MOT) for Unmanned Surface Vessels (USVs) using LiDAR data, several studies have proposed probabilistic point-based tracking-by-detection frameworks, which typically involve three key steps: data segmentation, data association, and Bayesian filter-based state estimation. The tracked objects are first detected, often through clustering algorithms like DBSCAN or Euclidean clustering [9]. Shape and pose estimation in point-based methods is generally performed on obtain clusters using L-shape fitting or ellipse fitting as done in [10], [11]. These fitting methods are typically formulated as optimization problems, where the objective is to minimize the sum of squared distances between the detected points and the proposed shape. This optimization is commonly solved using a least square fitting method proposed in [12]. While the size and shape estimation is decoupled from the tracking of the kinematic states in point-based approaches, extended object tracking (EOT) simultaneously tracks the position and shape of obstacles [13]. A master's thesis applied this theory to estimate vessel shapes using LiDAR data [14]. However, more recent research suggests that, until all measurement sources can be reliably modeled, point-based methods remain preferable [15].

Once detected, the tracks are updated by associating the predicted states with corresponding LiDAR cluster detections and then refining the predictions using those detections. In [16], a Global Nearest Neighbor (GNN) method is utilized to associate LiDAR detections with existing tracks, followed by the use of an Extended Kalman Filter (EKF) based on a Constant Velocity (CV) model for target tracking. Both [17] and [11] introduce an association algorithm that uses the Bhattacharyya distance, incorporating both position and size information for determining similarity. They employ the Iterative Closest Point (ICP) algorithm and a simple Kalman Filter (KF) for tracking, respectively. Another well-known tracking algorithm used for maritime settings is the Joint Probabilistic Data Association used in [18]. It is frequently highlighted in the literature as a potential solution for handling high-density clutter, and thus it is well suited in maritime environments where sea clutter can lead to high false alarm rates. In fact, research has shown that JPDA can be extended to explicitly model wake clutter within its framework, offering promising results for maritime applications. The Multiple Hypothesis Tracker (MHT), is also considered for LiDAR-based tracking. However, MHT is known to be complex to implement and computationally expensive, making it less suitable for scenarios with a high number of false alarms or objects [19]. Random Finite Set (RFS)-based approaches, such as the Poisson Multi-Bernoulli Mixture (PMBM) filter, represent another class of algorithms with significant potential [20]. PMBM has been established among the state-of-the-art methods in target tracking and recent research has explored the applicability of PMBM filters in maritime environments, notably for point target tracking [21]. This approach has been successfully demonstrated in closed-loop collision avoidance experiments, highlighting its feasibility for real-world applications. However, PMBM's inherent complexity, coupled with the absence of quantitative evidence on its inference times in real-time scenarios, raises concerns about its practical suitability for dynamic maritime tracking tasks, particularly in high-density point cloud environments such as coastal settings. Despite the various proposed frameworks, most existing studies presented above focus solely on tracking small boats and do not address the challenges of over-segmentation, which can arise when tracking larger vessels.

In nearshore environments, differentiating static obstacles from moving targets requires dynamic segmentation. Some studies achieve this by projecting the LiDAR point cloud onto a pre-existing map, where points that fall on static areas are filtered out [18], [22]. However, maps are not always available, and in such cases, simple velocity thresholding is often employed [16]. However, shadow motion, occlusion, jumping object frames, and noise can make velocity thresholding insufficient for performing dynamic classification [23]. Beyond velocity thresholding, [11] incorporates static mapping from unclustered points, with an inclusion check performed to classify a track as static if it remains within a mapped static region. Another study specifically focused on detecting and tracking riverbanks using a deep learning-based method [24].

Most existing work involving AIS data has been done in conjunction with marine radar [7]. In the literature, many studies focus on track-level fusion between radar and AIS measurements, also known as track-to-track fusion [25], [26]. In this approach, separate tracks are initiated and maintained for each sensor and then combined later at the fusion step. This method allows for more robust tracking by leveraging both radar's long-range detection and AIS's precise positional information. In contrast, AIS and LiDAR fusion has received

less attention, with the few existing studies primarily focusing on state estimation rather than addressing the complexities of data association [27], [28]. When data association is tackled, it is often in simplified scenarios where vessels are far apart, minimizing the potential for ambiguity in matching tracks from the two sensors [28], [29].

Building on insights from previous works, this study emphasizes the association of LiDAR and AIS data to mitigate the issue of oversegmentation caused by large vessels, while ensuring the LiDAR framework remains robust against vessel maneuvers and dynamic segmentation. Although wake clutter is a well-known challenge in maritime tracking, with methods proposed in [30] and [31], addressing this issue falls outside the scope of this work.

III. Methodology

A. Overview

The proposed framework consists of two main components: the LiDAR track, which processes LiDAR data, and the AIS track, which handles AIS data. These tracks follow a distributed late fusion paradigm to estimate the state of AIS-equipped vessels as shown in Fig.2. In the absence of AIS measurements, the framework relies solely on the LiDAR track, which follows a point-based tracking-by-detection approach. The detector component processes raw point cloud data, while the tracker maintains and updates detected objects' states. The framework is specifically designed to meet real-time requirements, ensuring that all processes run efficiently at or below the LiDAR update rate of 10 Hz for timely vessel tracking and state estimation.

B. LiDAR Object Detection

At each LiDAR sweep, a statistical noise removal filter is first applied to eliminate outlier points and reduce noise within the raw point cloud data. Additionally, points that are in close proximity to the ego vessel are removed using a distance threshold to prevent the inclusion of self-detection artifacts. To reduce the computational load, a voxel filter is applied to downsample the point cloud while preserving spatial coherence. Given that height information is unnecessary for tasks such as collision avoidance, and LiDAR measurements are sparse in the vertical direction, the point cloud is flattened onto a 2D plane by setting the z-component to zero. Then, an Euclidean cluster algorithm is applied to segment the point cloud into clusters, representing potential objects (vessels, buoys, river banks, etc). For each cluster, a 5-D detection vector $z_{LiDAR} = [x, y, \phi, l, w]$ is generated where x, y represents the cluster's centroid in an earth-fixed frame. The object's shape l, w, and its orientation ϕ are estimated using Principal Component Analysis (PCA) due to its computational efficiency and robustness, particularly in handling irregular point cloud distributions [32]. Unlike optimization-based methods, PCA is not prone to non-convergence, especially when the point cloud is unevenly distributed, making it challenging to fit a rectangle or ellipse accurately. Moreover, since the vessel's bridge is often detected, the centroid estimation provided by PCA remains sufficiently accurate for tracking purposes as shown in Fig1. In the absence of existing tracks, the detection vectors are used to initialize new ones.

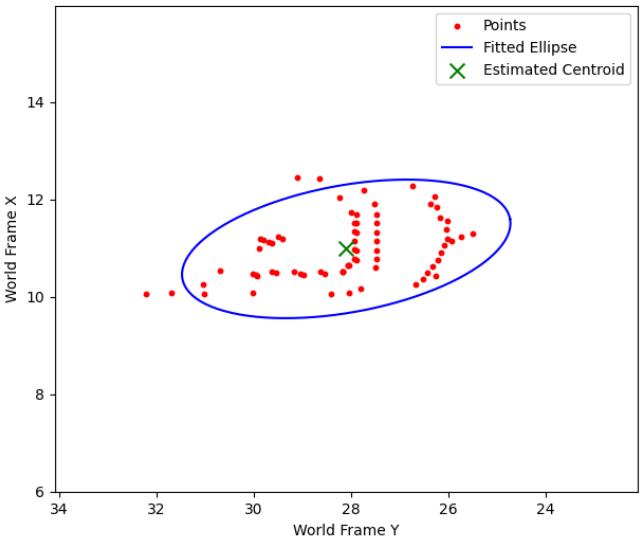


Fig. 1: Flattened point cloud of 9m long pleasure boat with Fitted ellipse and Centroid estimation using PCA.

C. LiDAR Multi-Object Tracking

Once the detection vectors are generated, the tracker component is responsible for maintaining and updating the state of these tracked objects over time, ensuring consistent and robust multi-object tracking in dynamic environments. A traditional probabilistic method relies mainly on data association and tracking filters. Additionally, a tracking management module serves to initialize, update, and delete tracks based on association history.

In this work, each track is characterized by the following state vector:

$$x = [p_x, p_y, v, \phi, \dot{\phi}] \tag{1}$$

where p_x and p_y represent the object's center position, v denotes its speed, and $\dot{\phi}$ represent the heading angle and rate of turn, respectively.

The data association problem is performed with the JPDA algorithm due to its ability to handle clutter and its potential to model wake clutter noise into its framework as discussed in Section II. The state filtering is performed using a non-linear Kalman-based filter, specifically the Extended Kalman Filter (EKF). Given the dynamic nature of vessel maneuvers, the Interacting Multiple Model (IMM) framework is used to switch between three motion models: Constant Velocity (CV), Constant Turn Rate and constant Velocity (CTRV), and Random Motion (RM). This allows the system to more robustly track objects with changing maneuvers and improves dynamic classification, especially in cases where static objects might mistakenly appear to be moving due to shadow motion. By incorporating motion model probabilities, the system reduces reliance on simple velocity thresholds. Tracks, where the probability of Random Motion is the highest among the three models, are more likely to represent static objects, providing more accurate classification of such objects and

improved robustness in LiDAR-based tracking. While similar frameworks, such as JPDA-IMM-UKF, have been applied in automotive [23], the JPDA-IMM-EKF has not yet been explored for maritime scenarios, which is a key contribution of this work.

To associate detections with existing tracks, a gating mechanism is applied to determine whether a detection is a valid candidate for a specific track. A detection is associated with a track q if it lies inside its elliptical validation gate $G(q)$. The validation gate of a track q is taken to be the same for all motion models where the validation gate corresponding to the motion model with the largest area is chosen:

$$G(q) = \{y_k = z_k : [z_k - \bar{\hat{z}}i_q, k]^T S^{-1}_{i_q,k}[z_k - \bar{\hat{z}}_{i_q,k}] \leq \gamma_G\} \quad (2)$$

where k is the time step, y_k is an associated detection, z_k is a detection, $S^{-1}_{i_q,k}$ is the inverse of the innovation covariance matrix at time step k with $i_q = \mathit{argmax}_{i \in M} det(S^i_k)$ being the index of the motion model in the model set and γ_G is the gate threshold. Multiple detections can be associated with a track if their Mahalanobis distance is smaller than a gate threshold obtained from the inverse chi-square cumulative distribution. Consequently, a validated set of associated detection is obtained for all models of a track q. Once detections are validated, the JPDA algorithm manages the assignment of multiple detections to a track by adjusting the Kalman gain. The state estimate is thus updated based on a weighted combination of the associated detections, allowing the system to handle ambiguous associations and improve robustness in cluttered environments. However, when a track is detected for the first time, the state estimate is initialized by interpolating between two time consecutive associated detections, selecting the detection with the smallest Mahalanobis distance from the validated set. This approach helps establish a well-initialized state (correct heading and speed). From the third time step onward, the update step is performed using the JPDA-IMM-EKF.

The filter struggles to accurately update velocity and angular velocity parameters, as LiDAR detections do not directly provide this information. Additionally, the accuracy of yaw estimation is affected by the degree of self-occlusion. To address these limitations, a weighted average of the previous five states is used to estimate velocity and angular velocity while the yaw history is integrated with the detected yaw into the detection vector. This approach ensures a more robust estimation and reduces the likelihood of unstable state updates.

$$\mathit{yaw}_{detector} = \alpha \times \mathit{yaw}_{detector} + (1 - \alpha) \times \mathit{yaw}_{history} \quad (3)$$

Inspired by [23], the tracking management module is responsible for handling track states and confidence levels based on detection associations. A confidence score between 0 and 1, which dynamically adjusts throughout the tracking process, is also used as a weighting factor in the computation of the Average Multi-Object Tracking Accuracy (AMOTA) and Precision (AMOTP), contributing to the overall performance evaluation of the system, as described in Section IV-B.

A track initially enters an Initialising state, where it must be consistently assigned detections for four consecutive time steps to transition to the Tracking state, with an increased confidence score. Tracks that fail this step are marked as Invalid. In the Tracking state, tracks continue to accumulate confidence when detections are assigned, while unassigned tracks move to a Drifting state, where their confidence decreases. If a track remains unassigned for a predefined number of time steps while drifting, it is invalidated. Invalid tracks are then removed from the track buffer.

Additionally, track pruning is applied to address the creation of duplicate tracks. The Euclidean distance between each pair of tracks is compared against a feasible distance for vessels in maritime environments. If the distance between two tracks is less than a specified threshold for more than five consecutive time steps, the newer track is deleted. This process mitigates the JPDA filter's tendency for coalescence, where neighboring tracks may be incorrectly associated with the same detection.

Lastly, tracks with a covariance matrix determinant exceeding a predefined threshold are considered unstable and are deleted to prevent coalescence. A large determinant indicates a broad uncertainty region, meaning the track's covariance matrix covers a large area, allowing many detections to be mistakenly associated with a single track during the gating step. By removing such tracks, the system avoids incorrect associations and improves tracking accuracy.

D. LiDAR-AIS Fusion

In addition to LiDAR tracks, this work integrates AIS data into the (MOT) framework by initializing tracks from AIS measurements. Since the framework is designed for near-range MOT, only AIS positions within a 150-meter radius of the ego vessel are initialized as new tracks. Each AIS message is composed of a following vector $z_{ais} = [x, y, CoG, \theta, SoG, ID, length, width]$, where SoG represents the speed over ground. For simplicity, the course over ground (CoG) and the heading θ are assumed to be identical in this work. Once both LiDAR and AIS tracks are established, the key task is determining whether a LiDAR track and an AIS track belong to the same vessel. By fusing the estimated states from both sources, AIS data can improve LiDAR-based state and shape estimation, while LiDAR can reduce the uncertainty in AIS tracks due to its higher update rate. To enable this distributed fusion, AIS tracks are synchronized with LiDAR timestamps, and their positions are adjusted based on the timestamp difference and reported velocity for proper alignment. Inspired by [28], AIS track states are updated directly based on their respective IDs using incoming AIS measurements, and an Extended Kalman Filter (EKF) with a Constant Velocity (CV) motion model is applied to these tracks. The final estimate is done through a weighted average of the respective estimates where the weights are derived from the Kalman gains, as shown in the following equations and illustrated in Fig.16.

$$W_{AIS} = K_{LiDAR,k+1} \left(K_{AIS,k+1} + K_{LiDAR,k+1}\right)^{-1}, \quad (4)$$

$$W_{LiDAR} = K_{AIS,k+1} \left(K_{AIS,k+1} + K_{LiDAR,k+1}\right)^{-1}, \quad (5)$$

$$x_{k+1} = W_{AIS}\ x_{AIS,k+1} + W_{LiDAR}\ x_{LiDAR,k+1} \quad (6)$$

Since the LiDAR track uses an Interacting Multiple Model (IMM) framework, the corresponding Kalman gain is computed as a weighted sum of the gains from the different motion models as follows

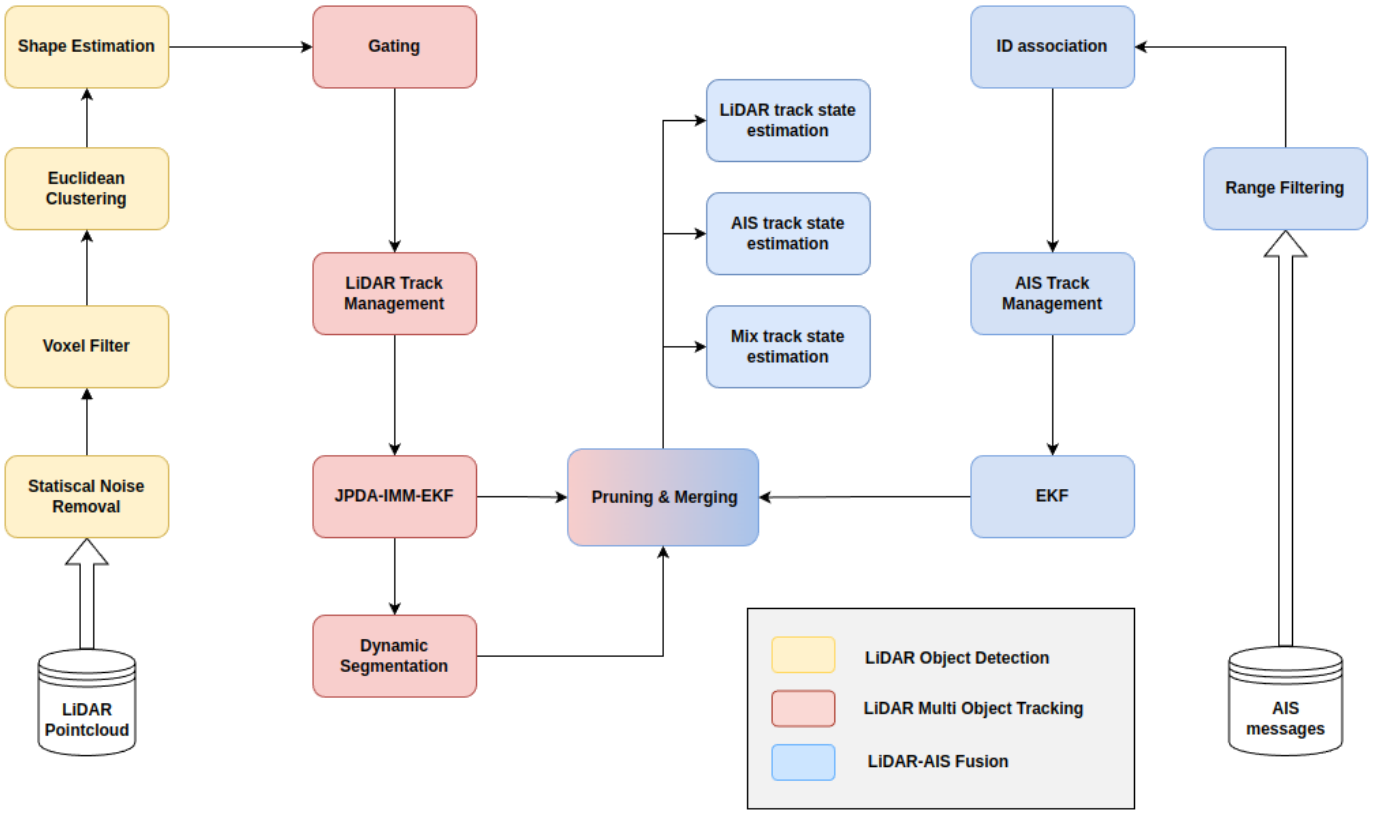


Fig. 2: Proposed MOT LiDAR-AIS Fusion Framework

$$K_{LiDAR} = K_{cv}\ \mu_{cv} + K_{ctrv}\ \mu_{ctrv} + K_{rm}\ \mu_{rm} \quad (7)$$

The equations above apply only when AIS and LiDAR measurements are assumed to originate from the same object. Therefore, this work focuses on effectively associating LiDAR tracks with their corresponding AIS tracks to ensure accurate fusion. To achieve this, an association method have been developed and is detailed below

***LiDAR-AIS Association Method**:* The association scheme is inspired by [27], where each non-static track is categorized as either a LiDAR, AIS, or Mix track. A Mix track incorporates both LiDAR and AIS data. For a track to transition into a Mix track, the fusion scheme addresses the assumption that, in the presence of large vessels, an Euclidean distance threshold may not be sufficient to fuse a LiDAR track with an AIS track. In this case, the size parameters provided by the AIS measurements are considered. Specifically, it checks whether the updated centroid of a LiDAR track falls within the ellipse defined by the length and width of the AIS track. The closest LiDAR track to the AIS centroid within this shape is fused with the AIS track to form a Mix track. The closest LiDAR track is then copied to the LiDAR component of the Mix track, before being deleted. Any remaining LiDAR tracks that also fall within the shape have their confidence score set to zero, ensuring they are not included in the evaluation process. The confidence score can be restored to its previous value if the LiDAR track ends up outside the AIS-defined shape.

At each time step, the LiDAR and AIS components of a Mix track are processed separately, and the final state estimate is computed as a weighted average of the two, as shown in Eq.6 when both components have associated detections. Until the next AIS update is received, the system updates only the LiDAR component, making the final estimate equivalent to the LiDAR state. If no detections are available for either component, the AIS prediction serves as the final estimate. For shape estimation, only the AIS data is used, as it is considered more accurate than the LiDAR detection. Lastly, if the LiDAR component of a Mix track becomes invalid due to consecutive unassociated frames, the track reverts to an AIS track, and the LiDAR data is no longer considered until a new fusion occurs, at which point the track transitions back to a Mix track.

IV. Experiments

A. Experiment details

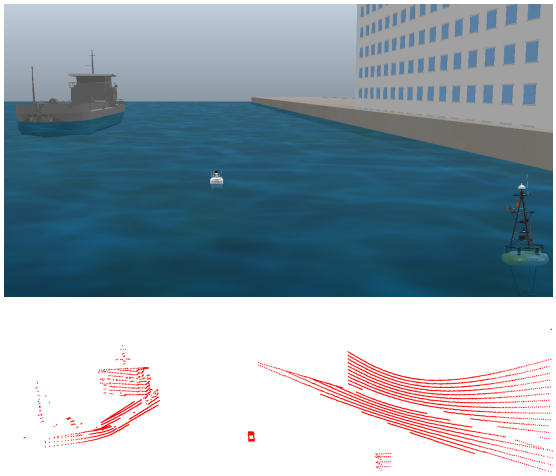


Fig. 3: Dynamic segmentation test with a hopper (grey) as target vessel, USV (white), buoy, river bank, and building; top: CoppeliaSim simulation, bottom: point cloud visualization

The proposed scheme was validated through a series of experiments conducted in the CoppeliaSim simulator. A custom LiDAR plugin within CoppeliaSim was used to simulate LiDAR data, with the LiDAR sensor mounted on the USV. The simulation environment is integrated with ROS2, which is used to publish sensor data, including LiDAR measurements and ground truth (GT) data, ensuring a realistic testing environment. To achieve fast inference time, the tracking node was implemented in C++.

The experiments involved four target vessels of different sizes, ranging from a 9m leisure boat to a 90m tanker, navigating around a stationary USV. Each vessel was tested across four main test types to evaluate tracking performance:

1) *Range Test*: The target vessel sails in a straight line perpendicular to the USV's long axis, with distances parametrized from 10m to 70m, testing the tracking accuracy over varying ranges.
2) *Self-Occlusion Test*: The vessel approaches or moves directly away from the USV along its longitudinal axis, assessing tracking performance under self-occlusion conditions.
3) *Proximity Test*: Two identical vessels sail parallel to each other and away from the USV along its longitudinal axis, with distances between them varied from 2m to 15m. This test evaluates the system's ability to distinguish closely spaced objects.
4) *Maneuver Test*: The vessel performs a 90-degree turn in front of the USV, challenging the framework's robustness in handling sharp directional changes.

Additionally, a separate *Dynamic Segmentation Test* was conducted to assess the framework's ability to differentiate between moving vessels and static structures. This test includes two scenarios:

1) *Static Obstacle Scenario*: Involves static obstacles only, including a floating cuboid, a 60-meter wall, and a buoy.
2) *Coastal Scenario*: Simulates a realistic coastal environment, featuring a moving vessel navigating near two buoys, a riverbank, and a building. This scenario assesses the framework's capability to distinguish between dynamic and static objects and evaluates its real-time performance under high point density conditions.

Each scenario excluding the static obstacle scenario was tested with and without AIS broadcasting. The AIS-LiDAR fusion system was evaluated in scenarios with AIS, while the LiDAR-only MOT framework was assessed without AIS. For each test, the target vessels travel 80 meters. The LiDAR-only MOT framework was initially tested with the 9-meter long vessel, traveling at three different speeds: 5, 10, and 15 knots.

Following this, the AIS-LiDAR fusion system was evaluated across all four vessels, but only at a speed of 10 knots, with AIS broadcasting enabled. To simulate real-world conditions, AIS messages were broadcast at random intervals between 1 and 10 seconds, with Gaussian noise added to the position and size. Additionally, the static ego vessel was subjected to sinusoidal rolling and pitching motions to simulate movement on water, further challenging the robustness of the tracking system.

Finally, a test set including two complex scenarios was designed. In the first scenario, all vessels were present simultaneously, leading to occlusion and varying numbers of vessels within the LiDAR's field of view. In the second scenario, a crossing maneuver between two vessels was performed in the presence of static obstacles such as a riverbank and buoys. More details on the experimental setup and parameters can be found in Appendix VII.

B. Experiment metrics

The performance of the proposed frameworks is primarily evaluated using two key metrics: Average Multi-Object Tracking Accuracy (AMOTA) and Average Multi-Object Tracking Precision (AMOTP). These metrics were initially developed for the nuScenes dataset and have become standard for evaluating multi-object tracking. AMOTA provides a summary of MOTA performance across various thresholds:

$$AMOTA = \frac{1}{n-1} \sum_{r \in \frac{1}{n-1}, \frac{2}{n-1}, \ldots, 1} MOTAR$$

$$MOTAR = \max(0,\ 1 - \frac{FP_r + FN_r + IDS_r + (1-r) \times TP}{r \times TP}) \quad (8)$$

where TP, FP, FN and IDS denote the true positive, false positive, false negative, and identity switch respectively. Traditionally, a match between ground truth (GT) and predictions is determined using Euclidean distance between their centroids. However, given the large size of the target vessels in this context, a distance threshold may not be optimal. Instead, a true positive (TP) is determined by an Intersection over Union (IoU) threshold of 0.3 between the GT and the prediction. Accordingly, AMOTP is defined as:

$$AMOTP = \frac{1}{n-1} \sum_{r \in \frac{1}{n-1}, \frac{2}{n-1}, \ldots, 1} \frac{\sum_{i,t} IoU_{i,t}}{\sum_t TP_t} \quad (9)$$

Both AMOTA and AMOTP take into account the confidence level of each detection.

Furthermore, for each true positive, the estimated length, width, and heading are compared against the ground truth. An average error is calculated across the entire scenario and recorded as the average length, width, and heading error. Lastly, the average inference time per scene is calculated to measure the real-time capability of the frameworks. Details on how these performance metrics are calculated after running the MOT framework on the recorded scenarios can be found in VII.

C. Run Time Parameters

For the JPDA-IMM-EKF framework used in this work, parameter values were adapted from an automotive tracking implementation to suit the unique dynamics and requirements of maritime vessel tracking. Notably, adjustments were made to manage the fusion of AIS and LiDAR data, with an emphasis on accounting for AIS's higher assumed precision. This is reflected in the noise matrices for each sensor: AIS measurements are assigned lower noise values than LiDAR to reflect their increased reliability. Regarding IMM, distinct process noise values for CV, CTRV and RM models are configured to better handle diverse vessel behaviors. Notably, the random motion model is assigned a higher noise level to accommodate unpredictable movements. An exhaustive list of parameters, covering all covariance, noise, clustering kernel size, and gating thresholds, can be found in Table XII.

V. Results

A. Experiments Results

The framework was evaluated across various scenarios with and without AIS broadcasting. Figure 4 presents the average AMOTA and AMOTP score for all scenes relative to vessel size, comparing results for both AIS-enabled and non-AIS setups. Additionally, Table III provides a detailed breakdown of MOT metrics for all vessels with AIS broadcasting enabled. Notably, the AMOTA scores for the 9m and 16m vessels are 0.69 $\pm$0.37 and 0.75 $\pm$0.34, respectively, indicating better tracking accuracy for smaller vessels. However, AMOTA and AMOTP drop significantly for larger vessels (50m and 90m) due to a large number of false positives rising to 90% and 81%, respectively. Additionally, the number of identity switches and track fragmentations increases with vessel size, complicating consistent tracking. For smaller vessels, the non-AIS setup performs better, as the AIS noise negatively impacts the tracking precision. Regarding the inference time, it increases gradually with the size of the target vessel; however, the framework remains within real-time requirements for simpler scenarios involving only moving vessels.

As a qualitative result, Figures 17 and 18 illustrate the effects of LiDAR-AIS fusion in a close proximity scenario involving two 9m vessels positioned 10m apart. Initially, the framework tracks both vessels separately using LiDAR data, assigning them IDs 0 and 1. Upon receiving AIS measurements, the system fuses these with LiDAR tracks, deleting the LiDAR-only tracks and creating Mix tracks. As the vessels move further away, tracking accuracy decreases due to a reduced point density, which introduces heading inaccuracies that may cause the estimated state to drift from the ground truth. Additionally it shows that the tracking management system also dynamically creates, deletes, and merges tracks as necessary, adapting to the scene. For instance, tracks 11 and 13 are generated, but some earlier tracks are removed or merged before their confidence score gets high enough to be plotted. This adaptability helps to reduce the number of unnecessary tracks, thereby lowering the number of false positives.

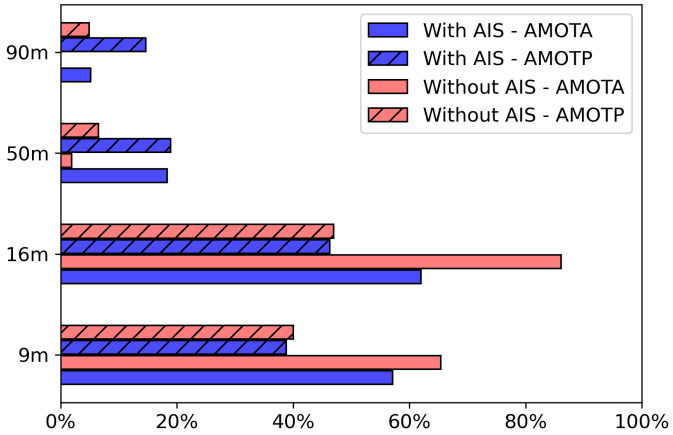


Fig. 4: Average AMOTA (no hatching) AMOTP (hatching) versus Vessel Length with AIS (blue) and without AIS (red).

B. Ablation Study

An ablation study was conducted to evaluate the proposed framework against variations without the IMM component and with a Nearest Neighbor data association approach instead of JPDA. This comparison aimed to assess the LiDAR framework's effectiveness across different test types and its capability to distinguish between static and moving obstacles through the IMM implementation. The three LiDAR framework variants were tested on all scenarios using the 9m target vessel without AIS broadcasting. Table III shows the average metric across all scenes excluding dynamic segmentation tests for the different LiDAR framework. Overall the 3 framework performs more or less similarly with amota of 0.64 $\pm$0.29 for NN, 0.73 $\pm$0.31 with JPDA and 0.69 $\pm$0.29 for IMM-JPDA. This is expected as the tracking are relatively simple and no clutter are present in the scenes. However, the slight decrease in performance for the NN framework may stem from its greedy data association step, which results in a higher number of false positives. This highlights the advantage of JPDA's probabilistic approach even without clutter, which provides more robust tracking when multiple detections of the same vessel appear, effectively reducing the likelihood of false positives. Table IV presents the tracking performance metrics per test type, highlighting the variability in accuracy across different scenarios. As expected, occlusion and maneuver tests exhibit the lowest performance metrics, with AMOTA values of 0.63 $\pm$0.31 and 0.55 $\pm$0.35, respectively. This decline is due

Vessel Size	AMOTA	AMOTP	Recall	TP(%)	FP(%)	FN(%)	IDS	Frag	LE	WE	IT(ms)
9m	0.69 ± 0.37	0.42 ± 0.12	0.87	72.60	20.88	26.93	1.0	2.0	0.48	0.59	4.47
16m	0.75 ± 0.34	0.48 ± 0.19	0.89	82.48	19.57	17.09	1.0	2.0	0.89	0.67	6.82
50m	0.18 ± 0.27	0.188 ± 0.10	0.48	40.18	90.02	59.61	10.0	6.66	0.80	0.69	14.35
90m	0.05 ± 0.10	0.15 ± 0.01	0.14	24.30	81.01	75.51	10.0	8.22	25.75	25.48	17.85

TABLE I: MOT Metrics per Target Vessel Size with AIS

Vessel Size	AMOTA	AMOTP	Recall	TP(%)	FP(%)	FN(%)	IDS	Frag	LE	WE	IT(ms)
9m	0.68 ± 0.27	0.42 ± 0.12	0.74	69.39	13.36	30.32	1.0	1.0	0.41	0.22	3.26
16m	0.87 ± 0.08	0.45 ± 0.08	0.96	93.48	7.79	6.25	1.0	1.0	1.58	0.43	4.50
50m	0.02 ± 0.06	0.06 ± 0.06	0.20	18.58	145.34	80.93	3.0	3.28	42.31	42.07	10.83
90m	0.0 ± 0.0	0.05 ± 0.07	0.14	16.27	197.64	82.85	3.0	6.20	60.98	58.65	15.18

TABLE II: MOT Metrics per Target Vessel Size without AIS

to the inherent challenges in these scenarios—occlusion reduces visibility, while sharp maneuvers create sudden changes in vessel dynamics, both of which strain the tracking framework's ability to maintain consistent associations.

The range yields comparatively better results. In particular, the range tests achieve an AMOTA of 0.853, benefiting from stable visibility and consistent trajectories that facilitate tracking. Close proximity tests, while more challenging than range tests due to closely spaced objects, still manage an AMOTA of 0.63 ± 0.31, demonstrating the framework's ability to handle scenes with multiple nearby targets. For a detailed look at the tracking performance across these scenarios, Fig. 19a and Fig. 19b illustrate the AMOTA scores for each LiDAR framework as a function of distance in the range tests and proximity between target vessels in the close proximity tests, respectively. For the range tests, the AMOTA scores remain around 0.8 up to a distance of 50m from the USV but decrease below 0.7 for the IMM-JPDA and JPDA frameworks and below 0.5 for the NN framework at distances beyond 50m. In the close proximity tests, performance is below 0.2 for all three frameworks when the vessels are only 2m apart but increases above 0.7 as the separation reaches 5m. Certain tests in both scenarios show high variability in results, indicated by larger standard deviations. This variability arises when the initial state estimation, based on interpolation between the first two frames, results in an estimated velocity opposite to the vessel's actual direction of travel. This issue is attributed to the USV's rolling and pitching motions, which alter the spatial distribution of LiDAR points and thus affect centroid estimation. Consequently, when velocity is estimated from consecutive centroids, the centroid in the current frame may appear behind the previous frame, giving the false impression that the vessel is moving in the opposite direction. This initial misalignment impacts both velocity and angle estimation, reducing overall tracking accuracy.

Table V presents the results of dynamic segmentation tests for tracking static obstacles in both scenarios described in Section IV. In the static obstacle scenario, the IMM-JPDA framework does not demonstrate improved performance, despite the expectation that the random motion probability would assist in segmenting dynamic obstacles amid random motion probability. Furthermore, the 60m-long wall alternates between being classified as static and dynamic due to substantial shadow motion. This effect is illustrated in Fig.20, where obstacles classified as static are marked in black. The wall can be observed transitioning between blue (dynamic) and black (static), highlighting the impact of shadow motion on the classification consistency. Additionally, all three frameworks exhibit a significant delay in recognizing static obstacles, as evidenced by a false negative rate exceeding 72%. The coastal scenario involves a moving vessel along a 150m-long riverbank. In this case, the river bank's size makes it challenging to successfully match with the ground truth, leading to an excess of false negative and ultimately resulting in poor tracking performance of the static obstacles. Similarly to the static scenario, the size of the river bank and the moving buoys prevent to constantly classify the river bank and moving buoys as static as shown in Fig. 21. Lastly, the average inference time increases significantly in this scenario, reaching approximately 98 ms for all three frameworks. This substantial increase is attributed to the large number of initialized tracks caused by the extensive riverbank.

C. Test Set

The framework was evaluated on a test set consisting of two complex scenes, as detailed in IV. The overall metrics for both dynamic and static obstacles in each scene are summarized in Table VI. The challenges observed in previous results are evident here as well. Specifically, the presence of large vessels and riverbanks generates a significant number of false positives, resulting in poor tracking performance for static obstacles. Additionally, the combination of large vessels, occlusions, and sharp maneuvers performed by other vessels in Scene 2 creates an exceedingly challenging scenario, leading to predictably poor tracking performance.

VI. Discussion

The results highlight key challenges and areas for improvement in the multi-object tracking framework, particularly linked to limitations in the detection component.

The experiments revealed a high rate of false positives for large vessels (50m and above), primarily due to the use of Euclidean clustering with a 3m kernel size. This clustering

LiDAR Framework	AMOTA	AMOTP	Recall	TP(%)	FP(%)	FN(%)	IDS	FRAG	LE	WE	IT(ms)
NN-EKF	0.64 ± 0.29	0.39 ± 0.13	0.70	62.54	31.36	36.69	1.0	5.0	0.31	0.18	4.26
JPDA-EKF	0.73 ± 0.31	0.44 ± 0.13	0.83	68.39	15.54	31.26	1.0	2.0	0.39	0.19	3.59
IMM-JPDA-EKF	0.69 ± 0.29	0.43 ± 0.13	0.81	65.06	15.15	34.45	1.0	1.5	0.37	0.22	4.46

TABLE III: LiDAR Frameworks

Test Type	AMOTA	AMOTP	Recall	TP(%)	FP(%)	FN(%)	IDS	FRAG	LE	WE	IT(ms)
range tests	0.85 ± 0.18	0.48 ± 0.08	0.90	80.28	3.86	19.49	0.0	0.0	0.18	0.21	4.04
maneuver tests	0.55 ± 0.35	0.34 ± 0.16	0.71	41.04	115.29	58.08	2.0	3.0	0.37	0.31	4.45
occlusion tests	0.63 ± 0.31	0.47 ± 0.13	0.83	66.20	9.77	33.19	1.0	1.0	1.85	0.24	3.27
close proximity tests	0.63 ± 0.31	0.40 ± 0.13	0.70	63.01	31.88	36.50	2.0	5.5	0.79	0.16	4.78

TABLE IV: Tracking Score of a 9m Long Vessel per Test Type

Dynamic Segmentation Tests	Lidar Framework	AMOTA	AMOTP	TP(%)	FP(%)	FN(%)	IT(ms)
Static Obstacles Scenario	NN-EKF	0.16	0.33	27.20	5.55	72.80	51.76
	JPDA-EKF	0.16	0.33	27.03	5.69	72.97	51.53
	IMM-JPDA-EKF	0.17	0.33	27.91	4.88	72.08	52.82
Coastal Scenario	NN-EKF	0.08	0.50	1.03	0.54	98.64	97.52
	JPDA-EKF	0.08	0.52	1.06	0.53	98.36	98.53
	IMM-JPDA-EKF	0.11	0.59	10.0	2.94	89.96	97.56

TABLE V: Tracking Score of Dynamic Segmentation Test

Test Set	Classes	AMOTA	AMOTP	Recall	GT	TP(%)	FP(%)	FN(%)	IDS	Frag	LE	WE	IT(ms)
Scene 1	**Vessels**	0.77	0.41	0.82	228	82.02	3.51	17.54	1.0	1.0	0.34	0.51	94.74
	Static Obstacles	0.0	0.0	0.0	114	0.0	458.77	100	NaN	NaN	NaN	NaN	94.74
Scene 2	**Vessels**	0.02	0.13	0.121	312	12.26	0.32	87.82	0.0	1.0	1.0	0.18	54.97
	Static Obstacles	0.0	0.0	0.0	156	0.0	0.0	100	NaN	NaN	NaN	NaN	54.97

TABLE VI: Tracking Score of Test Set

configuration generates excessive detections, leading to frequent track initializations. Despite the implementation of track management modules, the large size of vessels causes tracks to be too far apart for effective merging, complicating pruning and track management as shown in Fig.22. Similar issues arise with long static obstacles, such as riverbanks, where clustering inaccuracies persist. While AIS-LiDAR fusion marginally reduces false positives for larger vessels, the improvement is constrained by AIS's low update rate. For instance, in a 30-second scene, AIS data updates every 10 seconds, which is insufficient to significantly enhance tracking performance despite improved size estimation. Furthermore, the proposed fusion scheme associates the nearest LiDAR track within the AIS-defined shape to form a Mix track. Between AIS updates, the Mix track is solely updated by the LiDAR component, which often fails to represent the full extent of large vessels, resulting in reduced tracking accuracy. Adaptive clustering could address these issues by dynamically adjusting the kernel size based on AIS information, allowing clustering to match the expected number of vessels. This approach could enhance detection reliability, particularly for large vessels.

A critical challenge lies in incorrect initial state estimation, exacerbated by the rolling and pitching motions of the ego vessel. These movements distort the LiDAR point distribution, leading to errors in velocity and heading estimation during extrapolation between the first two detections. Once initialized, the IMM's Random Motion model and safeguard mechanisms can allow these erroneous tracks to persist, maintaining associations with detections despite incorrect motion dynamics and orientation. This results in tracks that continue to exist with unrealistic heading or velocity. A potential solution is to implement a MHT algorithm, which could maintain multiple hypotheses in the initial frames to improve the likelihood of correct heading and speed estimation. MHT could also dynamically adjust clustering parameters near known tracks, further mitigating clustering issues.

Tracking accuracy decreases as objects move further from the LiDAR sensor. To address this, the weighted fusion algorithm should consider object distance and orientation, assigning greater weight to AIS data for objects that are farther or poorly oriented relative to the USV. Additionally, incorporating perspective correction techniques, as described in [23], could improve tracking consistency during sharp maneuvers and occlusions.

Although motion compensation was applied, wave-induced rolling and pitching of the USV caused significant shadow motion. Mechanically stabilizing the LiDAR on a platform to maintain parallel alignment with the water surface could help maintain a stable point distribution across frames. This would facilitate more accurate IMM probability assessments, aiding in the classification of static and dynamic obstacles.

PCA is shown to be suboptimal for rectangular-shaped obstacles, such as those found among static objects. Incorporating an adaptive shape estimator that selects between L-shape fitting and PCA, as proposed in [10], could enhance accuracy. Additionally, the shape information obtained through this approach could serve as another cue for dynamically classifying obstacles. For instance, obstacles with rectangular

shapes are more likely to be static, providing valuable context for distinguishing between dynamic and static objects in the environment.

Practical challenges, such as issues with ROS 2 message handling, hindered the ability to automate sequential testing, requiring constant supervision for each test run. This significantly limited the opportunity to fine-tune parameters, as fine-tuning typically benefits from running a large number of scenarios to gather sufficient data for optimization. Additionally, the simulation environment did not simulate the effects of water on target vessels. For instance, the heading and course over ground were assumed to align, whereas real-world scenarios often see discrepancies caused by currents and wind. Waves further complicate tracking by introducing frame jumps and increasing shadow motion. Wake clutter, a significant challenge in maritime tracking was not addressed in this work due to the limited realism of the simulation environment. Moreover, the absence of noise in the simulated LiDAR point cloud data hindered the ability to demonstrate the advantages of JPDA over NN effectively. The coalescence problem, inherent to JPDA, was also not explored due to the lack of noise and the limited number of vessels in each scenario. Lastly, the stationary USV used in all experiments restricted the assessment of tracking performance under dynamic conditions.

Another point of discussion regards the precision metric used in this study. While AMOTP is typically expressed as an Euclidean error metric, this work opted to use IoU due to the challenges posed by large vessels, which often experience significant self-occlusion. Under such conditions, matching predictions with ground truth using Euclidean distance becomes unreliable. However, in retrospect, it might still be valuable to express AMOTP using Euclidean distance as it provides a more direct measure of localization error. The matching between predictions and ground truth could still rely on IoU, given its robustness to self-occlusion, while the Euclidean distance metric would better illustrate localization precision. Furthermore, as highlighted in [21], velocity and heading errors should also be included as evaluation metrics, as they hold greater significance for collision avoidance tasks. This aligns with the observation that collision avoidance systems primarily rely on predicted future positions, which are heavily influenced by accurate velocity and course estimates, rather than the current position. An inaccurate course estimate can therefore have a larger detrimental impact than an inaccurate position estimate, emphasizing the importance of these metrics in evaluating tracking performance.

VII. Conclusion

This work introduced a MOT framework for USVs that utilizes a late fusion of LiDAR and AIS data. The framework leverages probabilistic approaches such as the IMM-JPDA to manage dynamic vessel behaviors and potentially ensure robust data association in the presence of clutter. Alongside the framework, a set of experiments was designed and conducted using simulated scenarios to evaluate its performance. However, the experimental results revealed several limitations and areas for improvement. One key challenge lies in detecting large vessels and elongated static obstacles, such as riverbanks, which result in excessive false positives due to clustering inaccuracies. The fusion process also demonstrated shortcomings with large vessels, as it solely relies on the nearest LiDAR track within the AIS-defined shape to estimate the final state in the absence of AIS updates. However, this nearest track can still be far from the AIS centroid, causing the estimated LiDAR heading and position to deviate significantly from the true state of the target, leading to reduced precision. Additionally, the framework showed sensitivity to initial state estimation errors caused by the rolling and pitching motions of the USV. These movements distort the LiDAR point distribution, leading to inaccuracies in velocity and heading initialization. Future work should prioritize improving the detection component of the framework by implementing adaptive clustering strategies, enhancing track initialization, and developing more robust association mechanisms for AIS-LiDAR fusion. Additionally, incorporating a more realistic simulation of water dynamics—accounting for waves, wake clutter, and the effects of currents—along with introducing noise into simulated LiDAR data, is essential for meaningful validation of maritime tracking. Lastly, integrating the framework into a closed-loop collision avoidance system and combining it with occupancy grid mapping would represent a significant step toward practical deployment on real USVs.

Acknowledgments

I would like to thank Dr. H. Caesar and Ir. L. van Litsenburg for their guidance throughout this thesis. I am also grateful to Demcon Unmanned Systems for providing me the opportunity to work on this project. Thank you to my friends and family for their love and support, and to the Netherlands for providing me with an international engineering education.

References

[1] Z. Liu, Y. Zhang, X. Yu, and C. Yuan, "Unmanned surface vehicles: An overview of developments and challenges," *Annual Reviews in Control*, vol. 41, pp. 71–93, 2016. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1367578816300219

[2] H. Nguyen and B. Everett, "Joint robotics program (jrp)-supported efforts at the space and naval warfare systems center, san diego," *SPIE Unmanned Systems Technology VIII*, pp. 17–20, 05 2006.

[3] A. N. Cockcroft and J. N. F. Lameijer, *A Guide to the Collision Avoidance Rules: International Regulations for Preventing Collisions at Sea (the Seventh Version)*. Oxford: Butterworth-Heinemann, 2011.

[4] B.-n. Vo, M. Mallick, Y. Bar-shalom, S. Coraluppi, R. Osborne III, R. Mahler, and B.-t. Vo, *Multitarget Tracking*. John Wiley Sons, Ltd, 2015, pp. 1–15.

[5] S. Thombre, Z. Zhao, H. Ramm-Schmidt, J. M. Vallet García, T. Malkamäki, S. Nikolskiy, T. Hammarberg, H. Nuortie, M. Z. H. Bhuiyan, S. Särkkä, and V. V. Lehtola, "Sensors and ai techniques for situational awareness in autonomous ships: A review," *IEEE Transactions on Intelligent Transportation Systems*, vol. 23, no. 1, pp. 64–83, 2022.

[6] A. G. Bole, A. D. Wall, and A. Norris, *Radar and ARPA manual: radar, AIS and target tracking for marine radar users*. Butterworth-Heinemann, 2013.

[7] B. Habtemariam, R. Tharmarasa, M. McDonald, and T. Kirubarajan, "Measurement level ais/radar fusion," *Signal Processing*, vol. 106, pp. 348–357, 2015. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0165168414003636

[8] J. Lin, P. Diekmann, C.-E. Framing, R. Zweigel, and D. Abel, "Maritime environment perception based on deep learning," *IEEE Transactions on Intelligent Transportation Systems*, vol. 23, no. 9, pp. 15 487–15 497, 2022.
[9] M. Adnan, G. Slavic, D. Martin Gomez, L. Marcenaro, and C. Regazzoni, "Systematic and comprehensive review of clustering and multi-target tracking techniques for lidar point clouds in autonomous driving applications," *Sensors*, vol. 23, no. 13, 2023. [Online]. Available: https://www.mdpi.com/1424-8220/23/13/6119
[10] J. Lin, L. Koch, M. Kurowski, J.-J. Gehrt, D. Abel, and R. Zweigel, "Adaptive shape fitting for lidar object detection and tracking in maritime applications," in *International Journal of Transport Development and Integration*, 2021, pp. 105–117.
[11] Z. Yao, X. Chen, N. Xu, N. Gao, and M. Ge, "Lidar-based simultaneous multi-object tracking and static mapping in nearshore scenario," *Ocean Engineering*, vol. 272, p. 113939, 2023.
[12] A. Fitzgibbon, M. Pilu, and R. Fisher, "Direct least square fitting of ellipses," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 21, no. 5, pp. 476–480, 1999.
[13] K. Granstrom, M. Baum, and S. Reuter, "Extended object tracking: Introduction, overview and applications," 2017. [Online]. Available: https://arxiv.org/abs/1604.00970
[14] K. A. Ruud, E. F. Brekke, and J. Eidsvik, "Lidar extended object tracking of a maritime vessel using an ellipsoidal contour model," in *2018 Sensor Data Fusion: Trends, Solutions, Applications (SDF)*. IEEE, 2018, pp. 1–6.
[15] M. Baerveldt, A. G. Hem, and E. F. Brekke, "Comparing multiple extended object tracking with point based multi object tracking for lidar in a maritime context," *Journal of Physics: Conference Series*, vol. 2618, no. 1, p. 012011, oct 2023. [Online]. Available: https://dx.doi.org/10.1088/1742-6596/2618/1/012011
[16] J. Han, Y. Cho, J. Kim, J. Kim, N. Son, and S. Y. Kim, "Autonomous collision detection and avoidance for aragon usv: development and field tests," *Journal of Field Robotics*, vol. 37, pp. 987–1002, 2020.
[17] J. Lin, L. Koch, M. Kurowski, J.-J. Gehrt, D. Abel, and R. Zweigel, "Environment perception and object tracking for autonomous vehicles in a harbor scenario," in *2020 IEEE 23rd International Conference on Intelligent Transportation Systems (ITSC)*, 2020, pp. 1–6.
[18] Ø. K. Helgesen, K. Vasstein, E. F. Brekke, and A. Stahl, "Heterogeneous multi-sensor tracking for an autonomous surface vehicle in a littoral environment," *Ocean Engineering*, vol. 252, p. 111168, 2022.
[19] M. Motro and J. Ghosh, "Scaling data association for hypothesis-oriented mht," in *2019 22th International Conference on Information Fusion (FUSION)*, 2019, pp. 1–8.
[20] F. García-Fernández, J. L. Williams, K. Granström, and L. Svensson, "Poisson multi-bernoulli mixture filter: Direct derivation and implementation," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 54, no. 4, pp. 1883–1901, 2018.
[21] A. G. Hem, M. Baerveldt, and E. F. Brekke, "Pmbm filtering with fusion of target-provided and exteroceptive measurements: Applications to maritime point and extended object tracking," *IEEE Access*, vol. 12, pp. 55 404–55 423, 2024.
[22] W. Wang, B. Gheneti, L. A. Mateos, F. Duarte, C. Ratti, and D. Rus, "Roboat: An autonomous surface vehicle for urban waterways," in *2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2019, pp. 6340–6347.
[23] A. Kampker, M. Sefati, A. S. A. Rachman, K. Kreisköther, and P. Campoy, "Towards multi-object detection and tracking in urban scenario under uncertainties." in *VEHITS*, 2018, pp. 156–167.
[24] Y. Shan, X. Yao, H. Lin, X. Zou, and K. Huang, "Lidar-based stable navigable region detection for unmanned surface vehicles," *IEEE Transactions on Instrumentation and Measurement*, vol. 70, pp. 1–13, 2021.
[25] D. Gaglione, P. Braca, and G. Soldi, "Belief propagation based ais/radar data fusion for multi - target tracking," in *2018 21st International Conference on Information Fusion (FUSION)*, 2018, pp. 2143–2150.
[26] W. Chi Ming, L. Yanan, M. Lanxi, C. Jiuhu, L. Zhong, S. Sunxin, Z. Yuanchao, C. Qianying, C. Yugui, D. Xiaoxue, W. Jiayi, and Z. shunzhi, "Intelligent marine area supervision based on ais and radar fusion," *Ocean Engineering*, vol. 285, p. 115373, 2023. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0029801823017572
[27] J. Lin, A. Puthiyavinayagam, S. Liu, M. Kurowski, J.-J. Gehrt, R. Zweigel, and D. Abel, "Real-time multi-object tracking using adaptive filtering and filter banks for maritime applications," in *2021 European Control Conference (ECC)*. IEEE, 2021, pp. 2239–2244.
[28] D. Menges, A. von Brandis, and A. Rasheed, "Digital twin of autonomous surface vessels for safe maritime navigation enabled through predictive modeling and reinforcement learning," 07 2024.
[29] J. Han, Y. Cho, J. Kim, J. Kim, N. Son, and S. Y. Kim, "Autonomous collision detection and avoidance for aragon usv: development and field tests," *Journal of Field Robotics*, vol. 37, pp. 987–1002, 2020.
[30] A. Rodningsby, Y. Bar-Shalom, O. Hallingstad, and J. Glattetre, "Multitarget tracking in the presence of wakes," in *2008 11th International Conference on Information Fusion*. IEEE, 2008, pp. 1–8.
[31] E. Brekke, O. Hallingstad, and J. Glattetre, "Improved target tracking in the presence of wakes," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 48, no. 2, pp. 1005–1017, 2012.
[32] H. Zhao, Q. Zhang, M. Chiba, R. Shibasaki, J. Cui, and H. Zha, "Moving object classification using horizontal laser scan data," in *2009 IEEE International Conference on Robotics and Automation*, 2009, pp. 2424–2430.
[33] M. Sualeh and G.-W. Kim, "Dynamic multi-lidar based multiple object detection and tracking," *Sensors*, vol. 19, no. 6, 2019. [Online]. Available: https://www.mdpi.com/1424-8220/19/6/1474
[34] autoware Foundation, 2019. [Online]. Available: https://gitlab.com/autowarefoundation/autoware.ai/core_perception/
[35] nuTonomy, "nuscenes devkit," 2021. [Online]. Available: https://github.com/nutonomy/nuscenes-devkit

APPENDIX

Simulation

The simulated scenarios were developed using CoppeliaSim to evaluate the framework across a variety of conditions and vessel types. Movement control and data publishing within the simulation were handled using Lua scripts, with ROS2 serving as the middleware for communication. The paths followed by the vessels were generated through the CoppeliaSim API in Lua. A LiDAR plugin was integrated into the simulation, and AIS messages were generated using the CoppeliaSim API to retrieve the ground truth (GT) position. To simulate motion on water, the static USV is subject to sinusoidal rolling and pitching motions with magnitudes of $1/800sin(t)$ radians and $1/300sin(3t)$ radians, respectively.

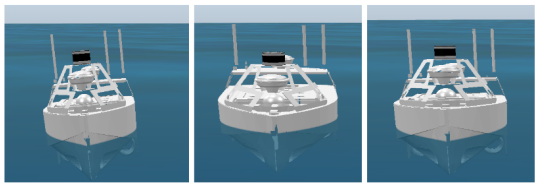


Fig. 5: USV with LiDAR mounted on top used in the simulation.

Four target vessels of varying sizes with distinct shapes were used in the simulation.

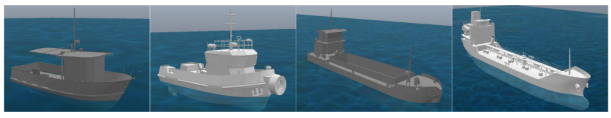


Fig. 6: Target vessels used in the experiments, from left to right (length x width): pleasure boat (9m x 3.2m), mooring boat (16m x 5.6m), hopper (50m x 10.5m), tanker (90m x 15.2m)

Four different type of tests were conducted to assess the tracking performance of moving vessels, focusing on aspects such as range, close proximity, occlusion, and maneuvering. Each test scenario was conducted at three different speeds—5, 10, and 15 knots. Additionally, dynamic segmentation tests

were performed separately to evaluate the framework's ability to distinguish between dynamic and static objects in the environment. All scenarios were captured as ROS bag files for further analysis. The following table, Tab.XIII, shows the number of LiDAR sweeps for all speeds and vessels, categorized by test type, with AIS broadcasting enabled.

Range test: The range test is designed to evaluate the framework's performance based on the distance between the target vessels and the USV. In this test, the target vessels sail in a straight line perpendicular to the USV. The distances range from 10m to 70m, with increments of 20m. An example is provided Fig.19a.

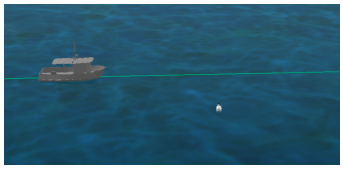


Fig. 7: Range test with a 9m long leisure boat as target vessel. In this example, the vessel's path is 10m away from the USV.

Self occlusion test: The self-occlusion test evaluates the framework's ability to track vessels when their orientation relative to the USV hinders accurate length estimation. In this test, the vessel moves directly toward or away from the USV along its longitudinal axis, as shown in Fig.8. As a result, only the vessel's width is detectable by the LiDAR.

Fig. 8: Self occlusion test where a 9m long leisure boat is sailing away from the USV.

Maneuver test: The maneuver test assesses the tracker's performance during a 90 degrees turning maneuver executed by a target vessel, as illustrated in 9.

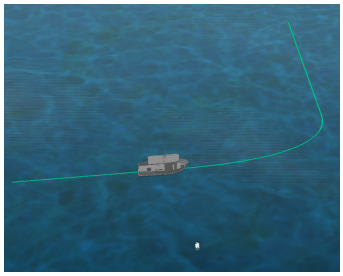


Fig. 9: Maneuver test with a 9m long leisure boat is performing a 90 degrees turn in front of the USV.

Proximity test: The proximity test involves two identical vessels sailing parallel to each other at a defined distance. This test evaluates the tracker's ability to handle closely spaced objects and identifies the distance at which it begins to misassociate targets due to proximity. An example featuring two tankers is shown in Fig.10.

Fig. 10: 2 90m long tankers at 10m from each other are sailing away from the USV in this close proximity test.

Dynamic segmentation test: The dynamic segmentation tests are divided into two distinct scenarios. The first scenario includes only static obstacles, featuring a floating cuboid, a 60-meter wall, and a buoy. The second scenario simulates a realistic coastal environment with a vessel navigating among two buoys, a riverbank, and a building, as shown in **??**. This test is designed to evaluate the framework's capability to differentiate between moving vessels and static infrastructure, as well as to assess its real-time performance given the high density of points generated in this complex setting.

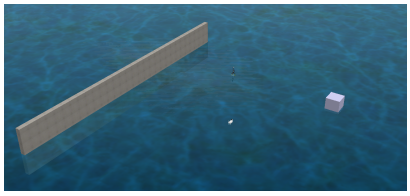


Fig. 11: Static Obstacles Only Scenario.

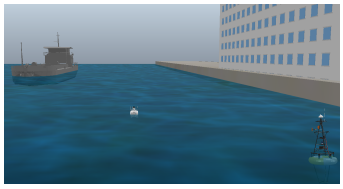


Fig. 12: Coastal Scenario where a 50m long hopper is sailing in the straight line. A building on a river bank is located on the right of the USV. 2 buoys are positioned behind the USV.

Test set: The test set consists of two scenarios. The first is characterized by two vessels performing a crossing maneuver near the USV and 130m long river bank.

Fig. 13: A crossing scenario between 2 vessels in front of the USV. A 130m long river bank is present behind the USV but can not be seen in the image.

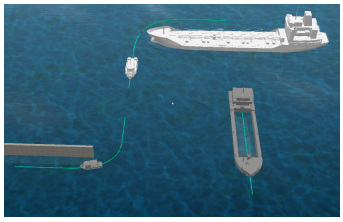


Fig. 14: A time-varying number of vessels within the LiDAR's field of view. 2 buoys and a wall serve as static obstacles.

The final scenario is a more complex scenario featuring a varying number of vessels within the LiDAR's field of view, along with buoys and a static wall, as depicted in 14.

Test type	Lidar Sweeps
range tests	10300
occlusion tests	3985
maneuver tests	4669
proximity tests	10437
dynamic segmentation tests	2048

TABLE VII: Number of LiDAR sweeps per test type.

Moving Vessel Test	GT
range tests	2668
occlusion tests	1645
maneuver tests	1262
proximity tests	5926
total	**11501**

TABLE VIII: Number of GT for LiDAR-only Tests.

Dynamic Segmentation Test	GT
Static Obstacles Scenario	498
Coastal Scenario	1844

TABLE IX: Number of GT for LiDAR-only Tests.

Moving Vessel Test	GT
range tests	1162
occlusion tests	668
maneuver tests	578
proximity tests	2724
total	**5172**

TABLE X: Number of GT per vessel for LiDAR-AIS Fusion Tests.

Test Set	LiDAR Sweeps
Scene 1	123
Scnene 2	86

TABLE XI: Number of LiDAR sweeps per scene in the test set.

Motion Models

CV model:

$$\begin{bmatrix} x_{t+1} \\ y_{t+1} \\ v_{t+1} \\ \theta_{t+1} \\ \dot{\theta}_{t+1} \end{bmatrix} = \begin{bmatrix} x_t + \Delta t \cdot v_t \cdot \cos(\theta_t) \\ y_t + \Delta t \cdot v_t \cdot \sin(\theta_t) \\ v_t \\ \theta_t \\ 0 \end{bmatrix} \tag{10}$$

CTRV model:

$$\begin{bmatrix} x_{t+1} \\ y_{t+1} \\ v_{t+1} \\ \theta_{t+1} \\ \dot{\theta}_{t+1} \end{bmatrix} = \begin{bmatrix} x_t - v_t \cdot \frac{\sin(\theta)}{\dot{\theta}} + v_t \cdot \frac{\sin(\Delta t \cdot \dot{\theta} + \theta)}{\dot{\theta}} \\ y_t + v_t \cdot \frac{\cos(\theta)}{\dot{\theta}} - v_t \cdot \frac{\cos(\Delta t \cdot \dot{\theta} + \theta)}{\dot{\theta}} \\ v_t \\ \theta_t + \dot{\theta}_t \cdot \Delta t \\ \dot{\theta}_t \end{bmatrix} \tag{11}$$

RM model:

$$\begin{bmatrix} x_{t+1} \\ y_{t+1} \\ v_{t+1} \\ \theta_{t+1} \\ \dot{\theta}_{t+1} \end{bmatrix} = \begin{bmatrix} x_t \\ y_t \\ 0 \\ \theta_t \\ 0 \end{bmatrix} \tag{12}$$

JPDA-IMM-EKF

The following set of equations describe the mathematical equation used in the JPDA-IMM-EKF. A thorough explanation of the equations are given in [33].

$$x_{k+1} = f_j(x_k, u_k) + w_{j,k}$$

$$z_k = h_j(x_k, u_k) + v_{j,k}$$

$$\Pi = \begin{pmatrix} p_{11} & \cdots & p_{r1} \\ \vdots & \ddots & \vdots \\ p_{1r} & \cdots & p_{rr} \end{pmatrix}$$

$$\mathbf{\Pi} = \begin{pmatrix} 0.9 & 0.05 & 0.05 \\ 0.05 & 0.9 & 0.05 \\ 0.05 & 0.05 & 0.9 \end{pmatrix}$$

$$\hat{x}^*_{j,k-1} = \sum_{i=1}^{r} \mu_{(i|j),k-1} \hat{x}_{i,k-1}$$

$$\hat{P}^*_{j,k-1} = \sum_{i=1}^{r} \mu_{(i|j),k-1} \left[P_{i,k-1} + (\hat{x}_{j,k-1} - \hat{x}^*_{j,k-1})(\hat{x}_{j,k-1} - \hat{x}^*_{j,k-1})^T \right]$$

$$\mu_{(i|j),k-1} = \frac{P_{ij}\mu_{i,k-1}}{\sum_{i=1}^{r} \hat{\mu}_{i,k-1}}$$

$$\beta^{cl}_{jq} = \sum_{\Theta} P\left\{\Theta | z_k\right\} \hat{w}_{jq}\left[\Theta\right], \quad j = 1, \cdots, N \ and \ \ q = 1, \cdots, T$$

$$P\left\{\Theta | z_k\right\} = \frac{1}{c} \prod_{j=1}^{N} g_{jq} P_D \prod_{q=1}^{T} (1 - P_D)^{\delta_q} \prod_{j=1}^{N} \beta^{\phi_j}$$

$$\tilde{z}_{i,q,k} = \sum_{j=1}^{N} \beta_{j,q} \tilde{z}_{i,j,q,k}$$

$$x_{i,q,(k|k)} = x_{i,q,(k|k-1)} + K_{i,q,k} \tilde{z}_{i,q,k}$$

$$P_{i,q,(k|k)} = P_{i,q,(k|k-1)} - \left(\sum_{j=1}^{N} \beta_{j,q} \right) K_{i,q,k} S_{i,q,k} K^T_{i,q,k} + K_{i,q,k} \left[\sum_{j=1}^{N} \beta_{j,q} \tilde{z}_{i,j,q,k} \tilde{z}^T_{i,j,q,k} - \tilde{z}_{i,q,k} \tilde{i,q,k}^T \right] K^T_{i,q,k}$$

$$\mu_{q,k} = \frac{\bar{\mu}_{q,k} \lambda_{j,k}}{\sum_{i=1}^{T} \bar{\mu}_{i,k} \lambda_{i,k}}$$

$$\hat{x}_{q,k} = \sum_{i} x_{i,q,k} \mu_{i,q,k} \quad \textit{alea iacta est}$$

$$P_{q,k} = \sum_{i} \left[P_{i,q,k} + \left(\hat{x}_{q,k} - x_{i,q,k} \right) \left(\hat{x}_{q,k} - x_{i,q,k} \right)^T \right] \mu_{i,q,k}$$

The noise covariance matrices for Lidar and AIS data represent the uncertainty in their respective measurements, with smaller values indicating higher confidence. For Lidar, it is assumed that equal uncertainty across x, y, and yaw dimensions, based on values from an open-source repository [34]. AIS data is considered more trustworthy, where the noise is set to 0.1. The final element, corresponding to the yaw rate, is set to 100 to reflect the absence of yaw rate measurements in AIS data; a placeholder value of 0 is used in the measurement vector, with the high variance indicating low confidence.

$$\mathbf{R\_lidar} = \begin{pmatrix} 0.25 & 0 & 0 \\ 0 & 0.25 & 0 \\ 0 & 0 & 0.25 \end{pmatrix}$$

$$\mathbf{R\_ais} = \begin{pmatrix} 0.01 & 0 & 0 & 0 & 0 \\ 0 & 0.01 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0.01 \\ 0 & 0 & 0 & 0 & 100 \end{pmatrix}$$

The process noise column vectors forming the process noise matrix for the state vector were inspired by a open source repository [34] and follow a general template used across motion models, where the corresponding parameters are replaced with specific values for different models; the template provided here is for the CTRV model.

$$\mathbf{q}_1 = \begin{pmatrix} \text{var\_ctrv} \cdot \text{var\_ctrv} \cdot dt^4 \cdot \cos^2(\text{yaw}) \cdot \text{ctrv\_var\_a} \\ \text{var\_ctrv}^2 \cdot dt^4 \cdot \cos(\text{yaw}) \cdot \sin(\text{yaw}) \cdot \text{ctrv\_var\_a} \\ \text{var\_ctrv} \cdot dt^3 \cdot \cos(\text{yaw}) \cdot \text{ctrv\_var\_a} \\ 0 \\ 0 \end{pmatrix}$$

$$\mathbf{q}_2 = \begin{pmatrix} \text{var\_ctrv}^2 \cdot dt^4 \cdot \cos(\text{yaw}) \cdot \sin(\text{yaw}) \cdot \text{ctrv\_var\_a} \\ \text{var\_ctrv}^2 \cdot dt^4 \cdot \sin^2(\text{yaw}) \cdot \text{ctrv\_var\_a} \\ \text{var\_ctrv} \cdot dt^3 \cdot \sin(\text{yaw}) \cdot \text{ctrv\_var\_a} \\ 0 \\ 0 \end{pmatrix}$$

$$\mathbf{q}_3 = \begin{pmatrix} \text{var\_ctrv} \cdot dt^3 \cdot \cos(\text{yaw}) \cdot \text{ctrv\_var\_a} \\ \text{var\_ctrv} \cdot dt^3 \cdot \sin(\text{yaw}) \cdot \text{ctrv\_var\_a} \\ dt^2 \cdot \text{ctrv\_var\_a} \\ 0 \\ 0 \end{pmatrix}$$

$$\mathbf{q}_4 = \begin{pmatrix} 0 \\ 0 \\ 0 \\ \text{var\_ctrv}^2 \cdot dt^4 \cdot \text{ctrv\_var\_yawdd} \\ \text{var\_ctrv} \cdot dt^3 \cdot \text{ctrv\_var\_yawdd} \end{pmatrix}$$

$$\mathbf{q}_5 = \begin{pmatrix} 0 \\ 0 \\ 0 \\ \text{var\_ctrv} \cdot dt^3 \cdot \text{ctrv\_var\_yawdd} \\ dt^2 \cdot \text{ctrv\_var\_yawdd} \end{pmatrix}$$

Performance Metrics Retrieval

The performance metrics are obtained using the nuScenes SDK [35], which provides a set of scripts to compute MOT metrics. These metrics are generated by reading a relational database containing predictions, annotations, and metadata. Each entry in the database as shown in Fig.15 is uniquely identified by a token and stored in a corresponding JSON file with relevant details. A dedicated ROS2 node was developed to subscribe to ground truth annotations and tracking predictions, automatically generating the required JSON files. As a result, for each rosbag run, a set of JSON files is created and updated after every callback.

The various nodes are managed by a Python script, which launches the appropriate files and plays the predefined rosbags. The number of rosbags to be tested can be specified in advance, with their paths listed in a text file. This file is then read by the Python script to initiate the different ROS2 nodes.

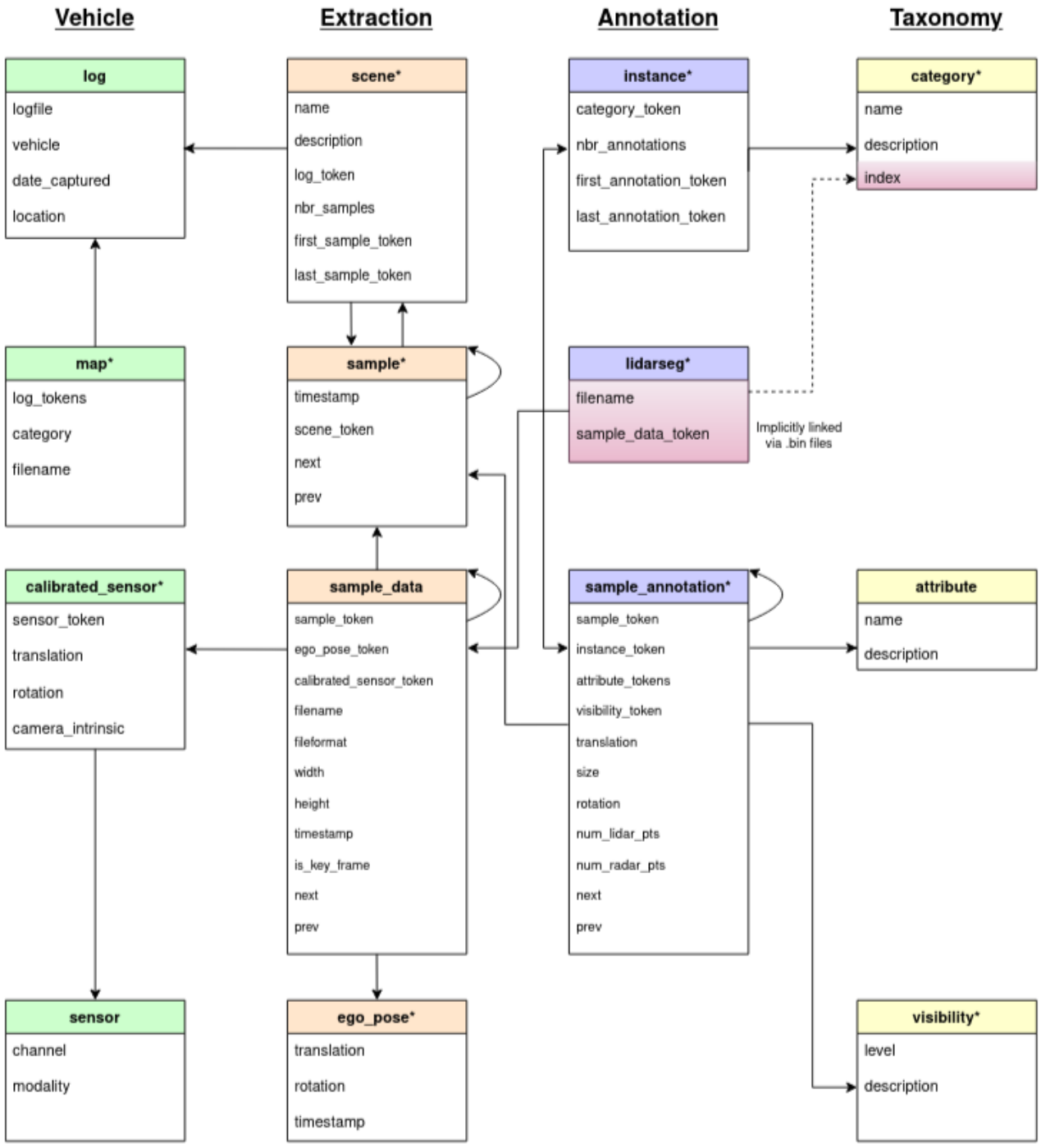


Fig. 15: Illustration of the relational database that must be created in order to run the nuscenes SDK [35].

Parameter	Value	Parameter	Value
gating_threshold	9.22	detection_probability	0.9
gate_probability	0.8	vel_change_threshold	6.0
yawd_change_threshold	0.5	decay	0.8
velocity_threshold	0.1	mode_prob_rm_threshold	0.35
max_ais_distance	130.0	cov_explosion_threshold	1000.0
amplifier	1.0	centerbox_size	1.0
leafsize	1.0	max_distance	60.0
cluster_tolerance	2.0	shape_dilatation	3.0
var_cv_lidar	0.5	var_ctrv_lidar	0.5
var_rm_lidar	0.5	std_a_cv_lidar	1.5
std_a_ctrv_lidar	1.5	std_a_rm_lidar	3.0
std_yawdd_cv_lidar	1.5	std_yawdd_ctrv_lidar	1.5
std_yawdd_rm_lidar	3.0	std_x_lidar	0.5
std_y_lidar	0.5	std_yaw_lidar	0.15
var_cv_ais	0.5	std_a_cv_ais	1.5
std_yawdd_cv_ais	1.5	std_x_ais	0.1
std_y_ais	0.1	std_v_ais	0.1
std_yaw_ais	0.1	std_yawd_ais	10.0

TABLE XII: Parameter Values for the IMM-JPDA-EKF Framework

Between each rosbag run, services are called to reset the TF buffer, data structures such as the track buffer, and callback inference time arrays.

To accommodate the maritime scenarios and specific metric requirements, the nuScenes SDK was modified accordingly. These adjustments included the addition of length and width average error calculations, the implementation of an IoU-based function to determine true positives (TP), and the introduction of a new category, namely ”vessel.”

When run, the tracking script outputs a set of metrics giving a thorough picture of the performance of the tracking framework over a scene. The following table lists the obtained metrics after running the metric code.

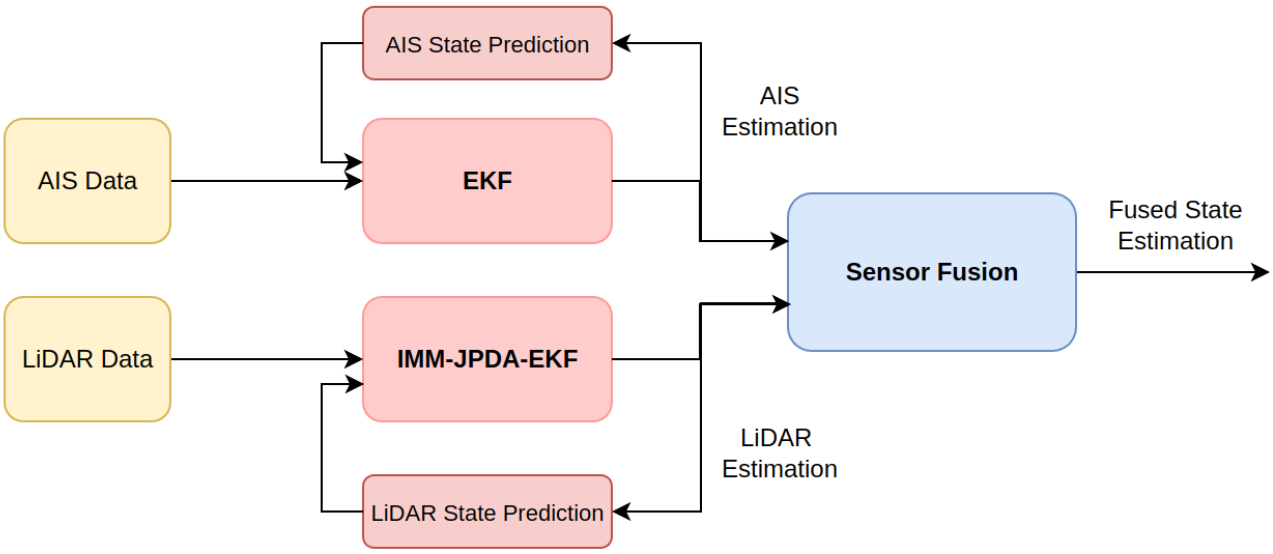


Fig. 16: Distributed Late Fusion Principle

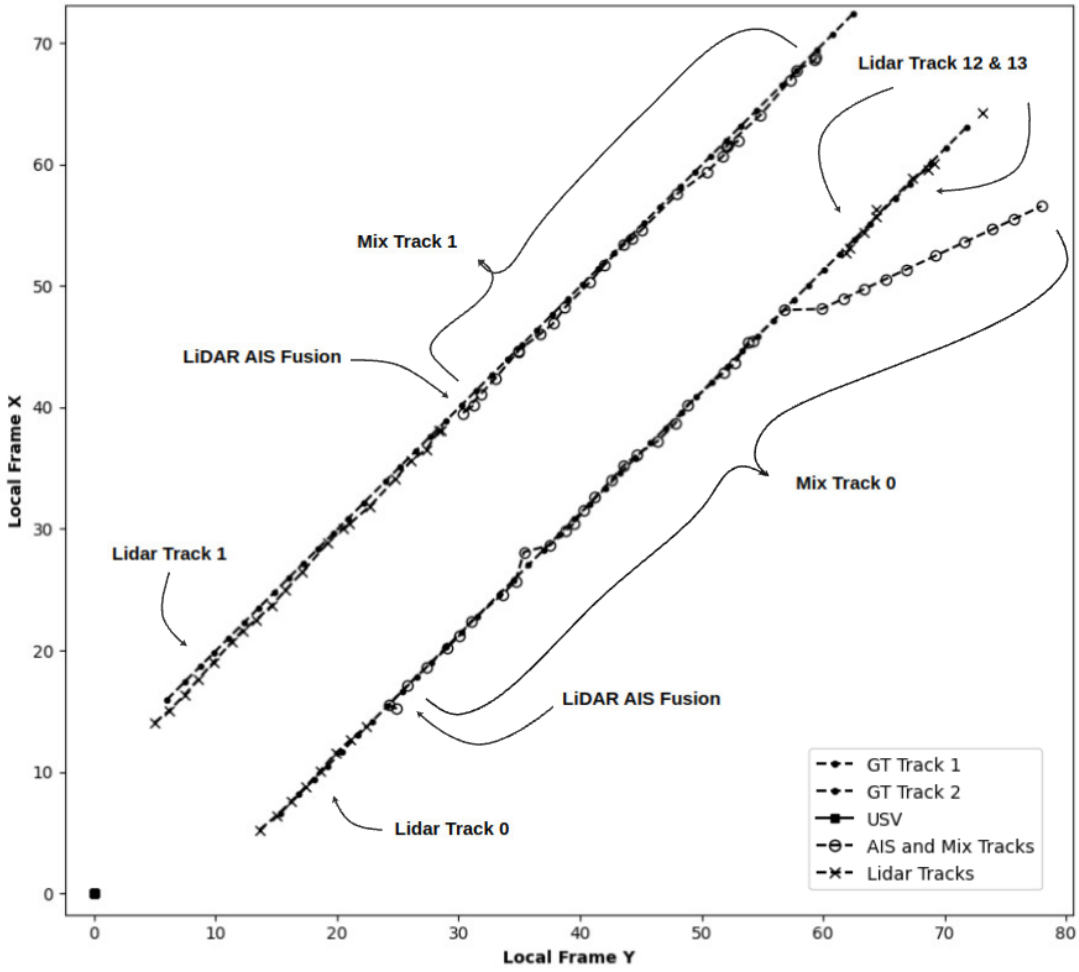


Fig. 17: Tracking Estimation along with Ground truth using LiDAR-AIS fusion framework for a close proximity test with 2 identical 9m long target vessels.

Metrics	Description
AMOTA	The overall tracking accuracy across multiple recall thresholds, combining false positives, false negatives, and identity switches.
AMOTP	The average localization precision of correctly matched objects, using Intersection over Union (IoU) across multiple recall thresholds.
Recall	The proportion of true positive out of all the ground truth objects in the scene
MOTAR	A metric that adjusts MOTA by taking into account the recall
MOTA	The tracking accuracy by considering false positives, false negatives, and identity switches
MOTP	The misalignement between the annotated and the predicted tracks
FAF	The average number of false alarms per frame
MT	The number of ground-truth trajectories that are covered by a track hypothesis for at least 80% of their respective life span
ML	The number of ground-truth trajectories that are covered by a track hypothesis for at most 20% of their respective life span
FP	The total number of false positives
FN	The total number of false negatives
IDS	The total number of identity switches
Frag	The total number of times a trajectory is fragmented
TID	Average track initialization duration in number of frames
LGD	Average longest gap duration in number of frames
LE	The average error in length estimation for all true positives
WE	The average error in width estimation for all true positives
IT	The inference time per frame average over the entire scene

TABLE XIII: nuscenes Metrics Description

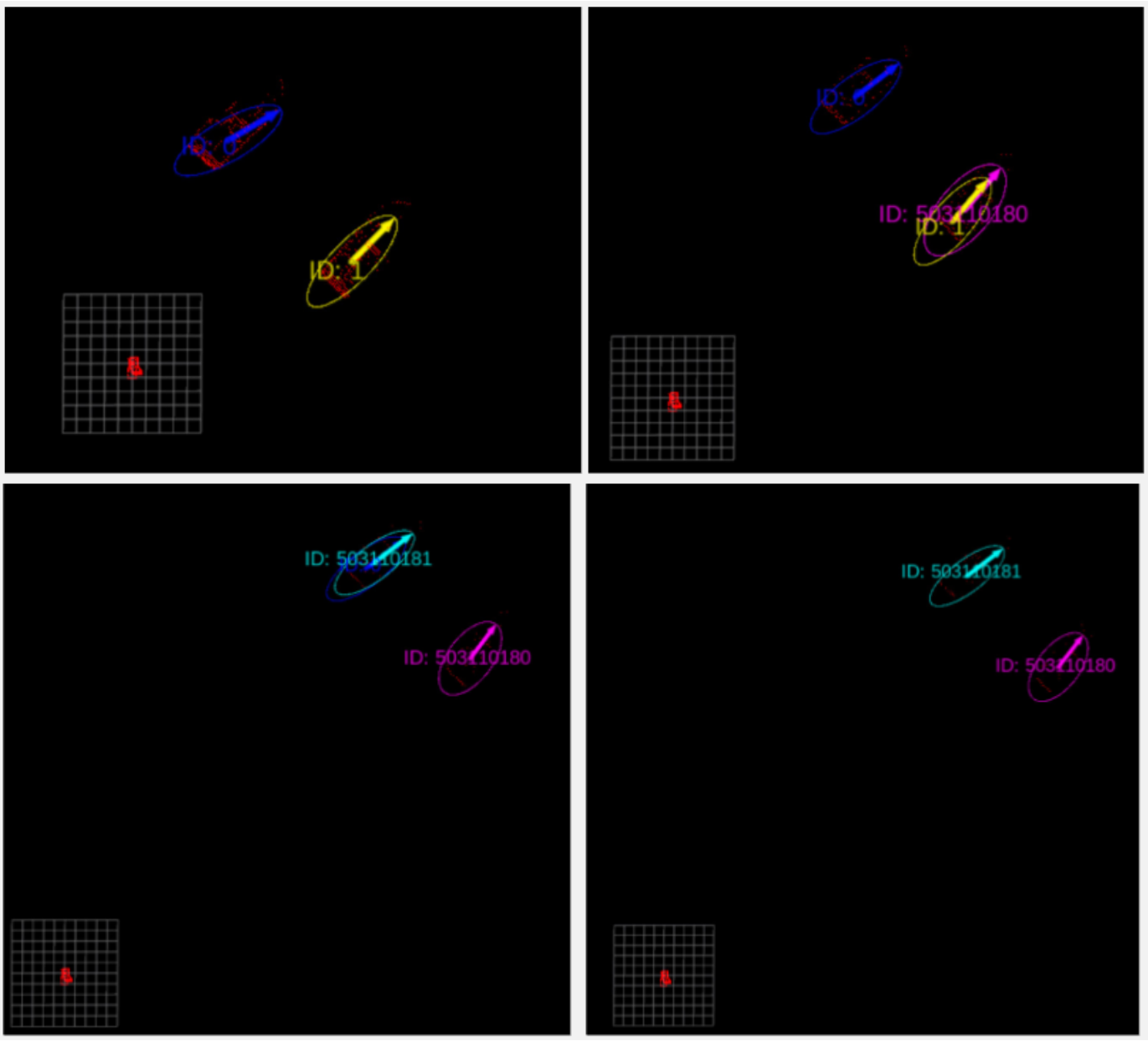
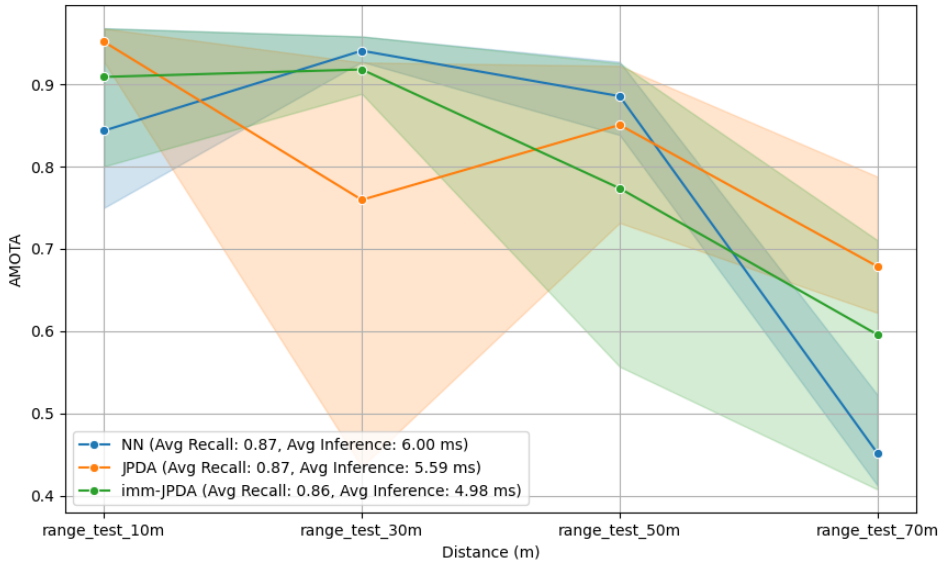
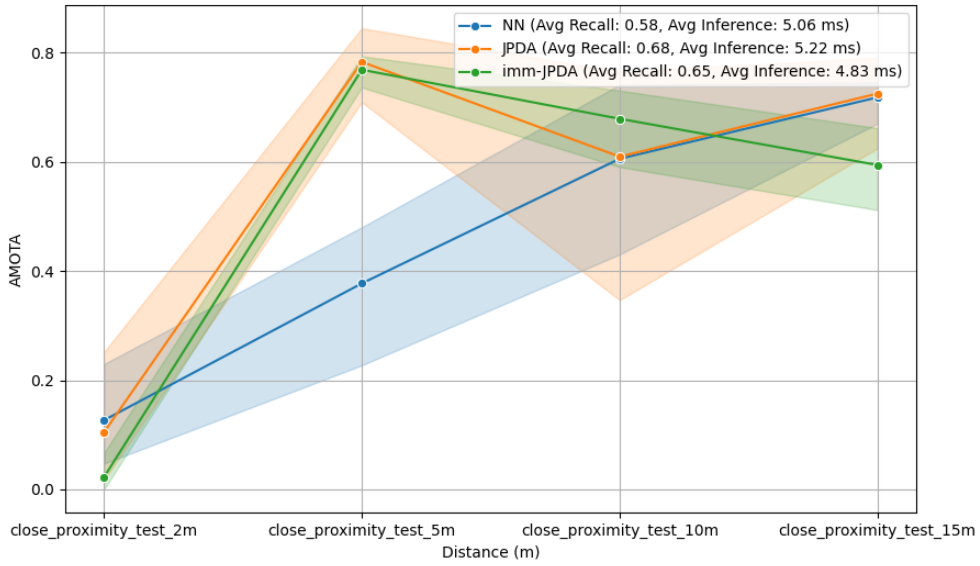


Fig. 18: Rviz visualization of a close proximity test with 2 identical 9m long target vessels. Red points are the raw Pointcloud. Each track is represented by an ellipse (shape), arrow (heading) and its ID. The USV is centered on the grid. 9 digits ID are ID provided by AIS messages.



(a)



(b)

Fig. 19: (a) AMOTA across Range Tests; (b) AMOTA across Close Proximity Tests

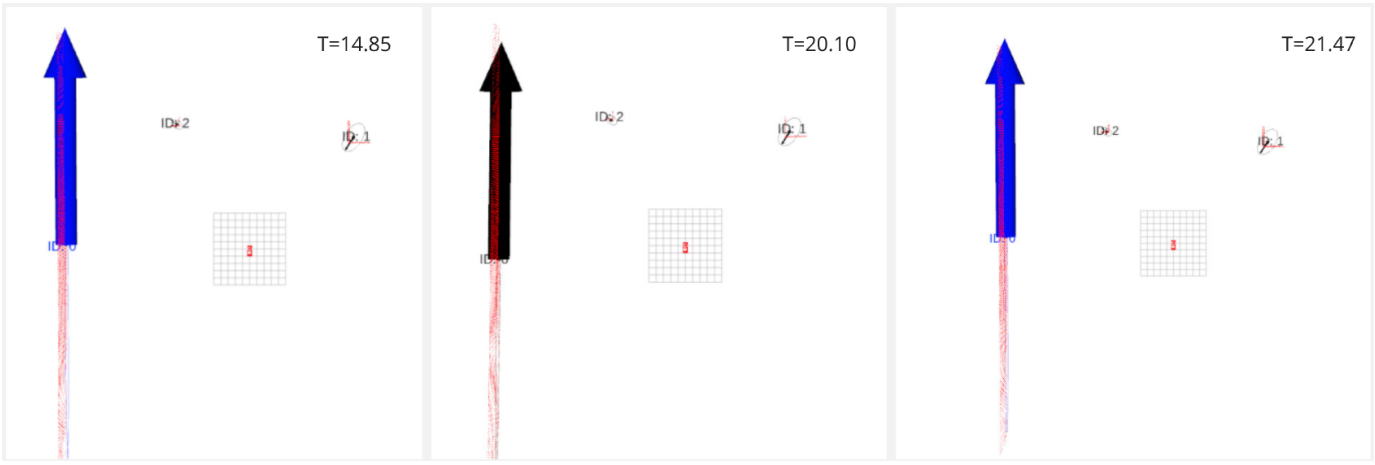


Fig. 20: Rviz visualization of Dynamic Segmentation (Static scenario) test. Red points are the raw Pointcloud. Each track is represented by an Ellipse (shape), arrow (heading) and its ID. Tracks classified as static are displayed in black. The USV is centered on the grid.

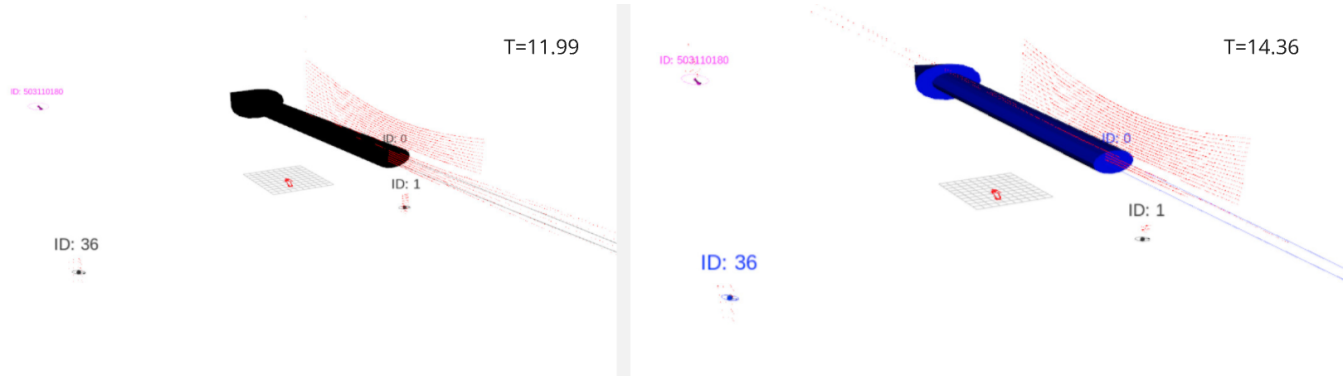


Fig. 21: Rviz visualization of Dynamic Segmentation (Coastal scenario) test. Red points are the raw Pointcloud. Each track is represented by an Ellipse (shape), arrow (heading) and its ID. Tracks classified as static are displayed in black. The USV is centered on the grid.

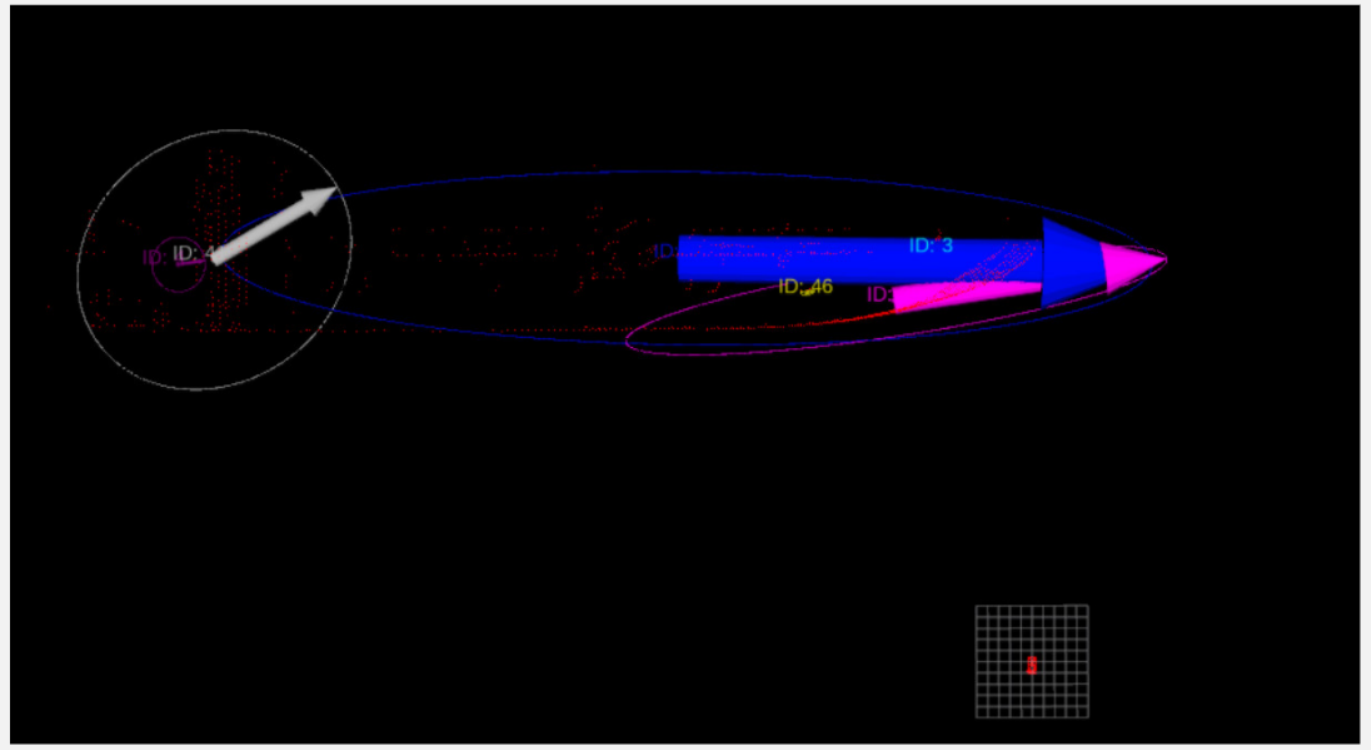


Fig. 22: Rviz visualization of Range Test with a 90m long target vessel. Red points are the raw Pointcloud. Each track is represented by an ellipse (shape), arrow (heading) and its ID. The USV is centered on the grid.