Maritime Tracking-By-Detection with Object Mask Depth Retrieval Through Stereo Vision and Lidar*

1st Henrik Hilmarsen† 2nd Nicholas Dalhaug† 3rd Trym Anthonsen Nygård† 4th Edmund Førland Brekke†

5th Rudolf Mester‡ 6th Annette Stahl†

† Department of Engineering Cybernetics, NTNU Trondheim, Norway

‡ Department of Computer Science, NTNU Trondheim, Norway

***Abstract*—The momentum towards autonomous technology is building up in the maritime domain, as the automotive industry has made big steps towards autonomous driving. The automotive industry has increasingly utilized visual methods for multi-object tracking (MOT), with the help of accessible benchmarking datasets such as KITTI. This paper presents a tracking pipeline that tracks in the world frame by using elements of a well-established visual tracking method that tracks objects in the image frame. The pipeline fuses 3D information from lidar or stereo vision with object masks from a deep learning-based ship detector. To handle occlusions, we implemented a track manager that predicts lost objects' movement until they reappear. Also, we provide a comparison between using lidar and stereo as the depth modality in the tracking pipeline. Results from a real-world experiment indicate that camera-lidar fusion gives consistently precise estimates, while the precision with stereo depends on the range and the type of vessel tracked.**

I. Introduction

Key components for ASVs include a control system, a navigation system, and a guidance system. The guidance system must be provided with situational awareness to effectively do path planning and collision avoidance [1]. Providing this awareness involves perceiving information about other moving objects in the environment of the vessel. Hence, the use of 3D MOT is crucial, as it significantly enhances the vessel's ability to navigate safely and efficiently in dynamic maritime environments.

Sensors used in maritime tracking include radar, lidar, optical cameras and infrared cameras. While radar is ubiquitous at long ranges, the higher resolution of cameras and lidar is required for precise situational awareness in harbors, canals and rivers. Optical cameras have been used in both monocular [2] and in stereo configuration [3]. An open research question is whether stereo cameras in the maritime domain can provide range data with an accuracy that for practical purposes is comparable to the accuracy of lidar.

The state-of-the-art approaches in visual tracking, popular in the automotive community, and in tracking using active sensors, which have dominated in marine autonomy, have evolved along different paths. In both schools, the pipeline typically consists of a detector, a solution for data association of the detections, and a solution for state estimation using the detections as data. In radar tracking, there has been a trend toward formulating major parts of this pipeline using Bayesian models. In contrast, visual tracking tends to rely less on models. There may be several reasons for this, including the following. First, visual tracking can utilize visual cues for association that make the Bayesian data association superfluous. Second, due to the lack of range information, tracking is typically done in the image frame, where it is difficult to make a meaningful target motion model.

* This work was supported by The Research Council of Norway (project number 333917).

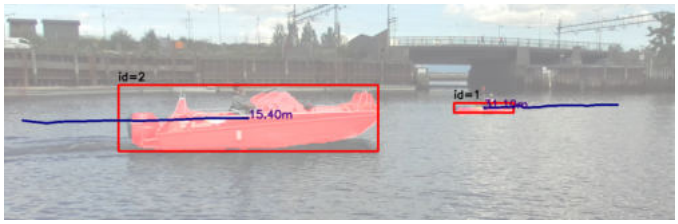

Fig. 1. Tracking visualized in the image coordinate system. The target boat and kayak are detected by YOLOv8 and tracked by BoT-SORT. The navy blue line illustrates the previous tracks of the object in the image coordinate system.

When designing a tracking system with cameras for use on ASVs, this leads to two options. On the one hand, one may use cameras as just another sensor in a multi-sensor fusion method, whose backbone could be a Bayesian tracking method such as the Joint Integrated Data Association (JIPDA) filter [4]. On the other hand, one may take a popular visual tracking method, such as Simple Online and Real-time Tracking (SORT) [5], as the starting point, and equip it with range information and occlusion handling. In this paper, we follow this second approach, which to the best of our knowledge has not been pursued previously in the maritime domain.

The contributions of this paper can be summarized as follows.

- We develop a tracking pipeline for autonomous surface vehicles that utilize either a camera-lidar fusion, or just a stereo camera. These two configurations are named: Visual Lidar Multi-Object Tracker (VLMOT) and Visual Stereo Multi-Object Tracker (VSMOT).
- We demonstrate the performance of the tracking pipeline by comparing it with global navigation satellite system (GNSS) data for real targets.
- We provide a comparison between using lidar and stereo camera as the depth modality for the tracking pipeline.

This paper uses the same data as a similar paper [6], which investigates using a multi-baseline stereo setup paired with JIPDA for tracking.

II. Related Work

Tracking-by-detection is its own paradigm within object tracking. A tracking-by-detection-based tracking scheme first detects objects and then attempts to associate these with existing tracks [7]. Thus, tracking-by-detection schemes are very dependent on the quality of the detections. At sea, acquiring good detections may be challenging for image object detectors due to harsh weather conditions, challenging lightning, or camera motion violating the common static camera assumption.

SORT [5] has become a family of trackers, and is a well-known tracking-by-detection tracker. SORT itself uses bounding boxes from a detector and associates them frame-to-frame using the Hungarian algorithm and a Kalman filter (KF) for each track. The tracking is solely based on image data and relies only on motion cues for association. SORT is easy to extend, which is why there are many different variants of it, e.g., DeepSORT [8], StrongSORT [9], OC-SORT [10], Deep OC-SORT [11], ByteTrack [12] and BoT-SORT [13]. Bag-of-Tricks SORT's (BoT-SORT) SOTA performance on MOT17 and MOT20 and its ease of use made it a suitable choice for our pipeline. BoT-SORT's main improvements from SORT is a second data association with low score detections, camera motion compensation, and a slightly different KF.

Because SORT relies solely on its detections and the intersection-over-union (IoU) between predicted and detected bounding boxes, it is prone to ID switches. Lack of detections, camera motion, variation in bounding box size, etc. may lead to a low IoU score and, therefore, ID switches. Therefore, we have implemented a track manager on top of BoT-SORT, which matches lost tracks with newly and erroneously created tracks using 3D information.

For tracking in 3D, fusing camera and lidar data have become widely adopted, especially in the automotive industry. A state-of-the-art method within 3D tracking is EagerMOT [14]. They detect objects in both lidar scans and monocular images and then fuse measurements in the image domain. They use both appearance and motion in the data association to get tracks. Note that the detection in both the lidar scans and the camera images requires a learned network. A SOTA method on KITTI [15] using camera and lidar is VirConvTrack [16]. The detector provides 3D detections of objects from point clouds formed by image and lidar data. The tracker employs a constant velocity model together with a KF and prediction confidence for data association [16]. EZFusion [17] is another tracking framework fusing camera, lidar and radar data and achieves top performance on nuScenes [18]. The results of these papers support the implementation of a camera-lidar fusion for tracking in the maritime domain, as we have done in this paper.

In the maritime domain, pioneering ASV projects like [19] perform 3D tracking by fusing lidar, radar, stereo and AIS data. Furthermore, [20] and [21] employ stereo vision for object detection and tracking.

A similar pipeline to ours is developed in [22], which exploits a camera-lidar fusion for tracking. They follow a SORT-like tracking scheme which tracks in the image frame. Opposed to our approach, which projects the lidar points into the image, they project the 2D object bounding box into the 3D lidar point cloud. The bounding box projection forms a volume used to filter out all lidar points not within the volume. Then, the closest cluster of lidar points within this volume is considered the detection, which makes the method very vulnerable to occlusions. Contrarily, our approach is capable of handling occlusions.

In a recent paper, [23] from 2023, a lidar-based MOT scheme that employs DBSCAN for detections and a deterministic tracking scheme is introduced. In [24], the work from [23] is extended to fuse camera and lidar. Tracking is still done only for the lidar coordinate system, but the tracked clusters are projected onto the image coordinate system and associated with bounding boxes provided by You Only Look Once version 8 (YOLOv8). However, this does not improve the tracking performance as the image detections are not directly utilized for tracking, like in our pipeline. Another recent paper [25] fuses camera, lidar and radar information for tracking in a similar manner. However, these papers do not appear to deal with occlusion or ID switches, which our tracking pipeline does by predicting lost objects' movement until they reappear.

III. Experimental Setup and Data

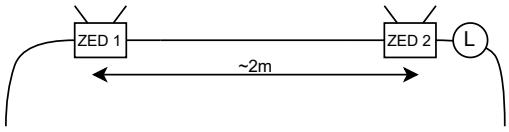

Fig. 2. The setup on the aft side of milliAmpere 2. The ZED 1 and ZED 2 cameras are placed about 2 m a part. The lidar sensor is denoted by "L", close to the ZED 2 camera.

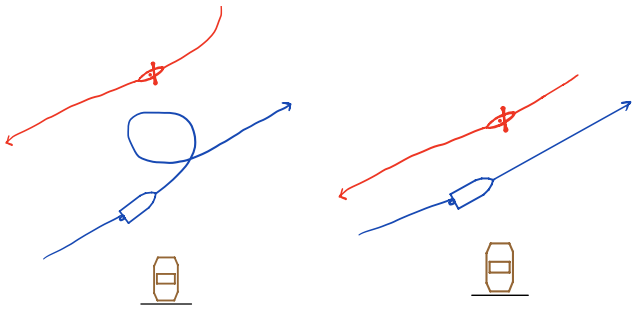

Fig. 3. Hand-drawn boat and kayak trajectories for test episodes 1 (a) and 2 (b). Kayak in red, boat in blue, and MA2 in brown.

The data used in this project is collected at Ravnkloa in Trondheim, Norway using the milliAmpere 2 (MA2) ferry

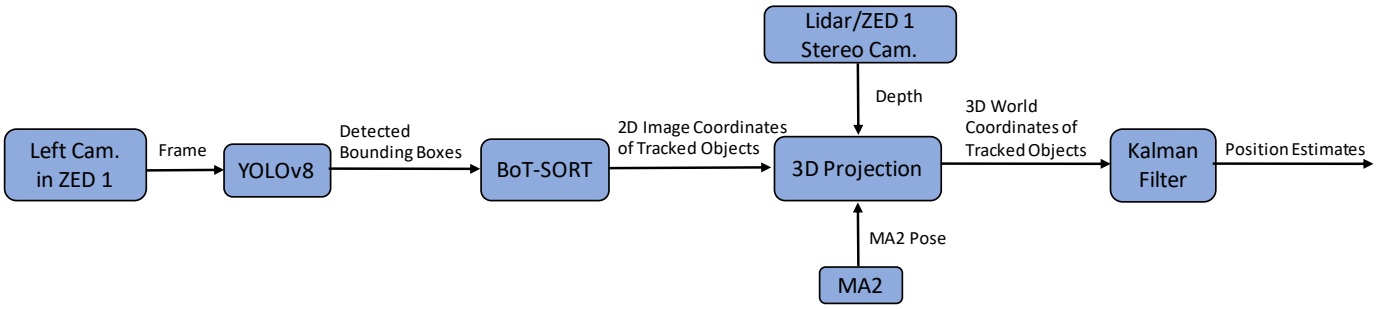

Fig. 4. The flow of the tracking pipeline.

Fig. 5. The target kayak (a) and Buster XL boat (b).

[26]. The dataset contains video, lidar, GNSS, and inertial navigation system (INS) data from MA2 in different settings. The cameras used are a ZED 1 and a ZED 2 camera. For this paper, the data from the ZED 1 stereo camera and an Ouster OS1-32 lidar sensor mounted on MA2 are primarily used. The setup of the cameras and lidar is illustrated in figure 2. In addition, the INS data from MA2 is utilized in the pipeline. The INS consists of an ADIS16495 inertial measurement unit (IMU) and a uBlox F9P Dual GNSS board. The dataset also contains GNSS data for the targets in the different episodes. The targets in the dataset are a Buster XL boat and a kayak. Images of the boat and kayak are shown in figure 5. The two episodes used in this project are illustrated in figure 3, named episodes 1 and 2.

IV. Tracking Pipeline

The proposed tracking pipeline is illustrated in figure 4. YOLOv8 [27] is used for object detection and BoT-SORT [13] is used for object tracking in 2D. The Ultralytics [27] BoT-SORT implementation without the re-identification module is used in this paper. Furthermore, the depth is measured by either a stereo camera or a lidar sensor. Using the 2D coordinates of the object, the depth and the pose of MA2, the world coordinates of the object can be calculated. Afterward, the 3D positions for each object are Kalman-filtered for smoothed estimates and prediction capabilities. Additionally, a track management module keeps track of lost tracks and matches reappearing objects to lost tracks.

A. Object Detection and Tracking in 2D

YOLOv8, specifically the pre-trained segmentation model trained on COCO [28], is the object detector used in the tracking pipeline. YOLOv8 provides object masks and object bounding boxes with center coordinates, width, and height. Figure 1 illustrates the bounding boxes and the object masks.

BoT-SORT is the 2D object tracker used in the tracking pipeline, which however is replaceable. As mentioned, BoT-SORT is an improvement of the well-known SORT and is further explained in [13]. From BoT-SORT we get the object mask, ID and 2D coordinates for all tracked objects which we use further in the pipeline. In figure 1, BoT-SORT is paired with YOLOv8 for tracking. The kayak is given ID 1 and the boat ID 2.

B. Depth Measurement

The next step in the tracking pipeline is to retrieve the depth of the tracked object to calculate the 3D position. Regardless of using stereo or lidar as the depth sensor, a depth map is obtained for every frame, either from the stereo camera or created for lidar as described below. The depth map is of equal size as the image and only has depth values on the pixels where a depth measurement was obtained. This is because the stereo camera has some blind spots and far-away points where the depth is not measured, and the lidar sensor provides sparse depth measurements.

In the VSMOT configuration, a stereo camera is used as the depth sensor, specifically a ZED 1 camera. The ZED camera is precalibrated and automatically performs image rectification, disparity, and depth calculations for each frame. The depth map is the same size as the image, with depth values only on pixels where depth measurements were obtained. Pixels with invalid depth are set to have $0\,\mathrm{m}$ depth. Such a depth map is visualized in figure 6a.

In the VLMOT configuration, a lidar sensor is used to measure the depth. At each image frame, there is a corresponding lidar point cloud. The lidar points are projected onto the image to retrieve the depth of the object, effectively forming a lidar depth map. The projection of lidar points onto the image coordinate system can be explained as follows. Let $\mathbf{x}^r$ be a 4×1 homogeneous point vector from the lidar point cloud, where $\{r\}$ denotes the lidar coordinate system. Furthermore, we let $\{c\}$ denote the stereo camera coordinate system. Let

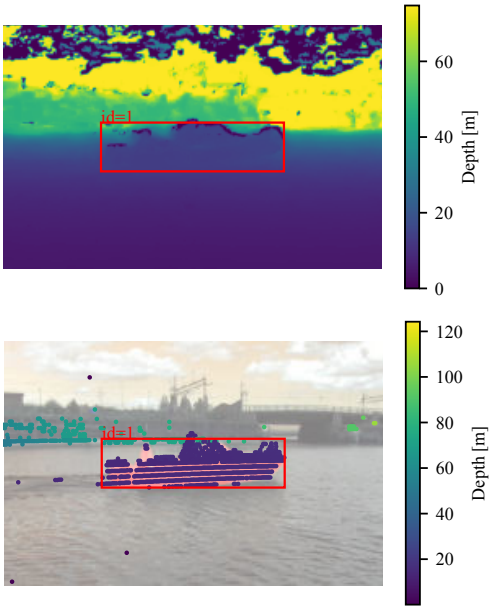

Fig. 6. Stereo depth map (a) and lidar depth map (b) with red bounding box provided by YOLOv8 and BoT-SORT tracking target boat with ID 1. The color represents the depth at the given pixel. The lidar dots are enlarged for visualization purposes. In reality, each lidar dot correspond to a single pixel.

$\mathbf{H}_r^c$ be the 4×4 homogeneous transformation from $\{r\}$ to $\{c\}$. The point $\mathbf{x}^r$ is transformed to $\{c\}$ by

$$\mathbf{x}^c = \mathbf{H}_r^c \mathbf{x}^r \tag{1}$$

where $\mathbf{x}^c$ is the corresponding 4×1 homogeneous point in the camera coordinate system. Because the lidar sensor measures depths in all directions and the camera is pointed along the positive depth direction, only positive depths are considered. The camera point $\mathbf{x}^c$ can be projected onto the 2D image coordinate system as $\mathbf{x}^{\text{im}} = \mathbf{K} \begin{bmatrix} \mathbf{I}_{3\times 3} & | & \mathbf{0}_{3\times 1} \end{bmatrix} \mathbf{x}^c$, where $\mathbf{K}$ is the camera calibration matrix for the stereo camera and the subscript $\{\text{im}\}$ denotes the 2D image coordinate system. The lidar points outside of the image boundaries are excluded, leading to a depth map that has the same dimensions as the image. In this map, all lidar points present in the image scene will be assigned to a pixel. Figure 6b shows an example of the lidar projection, where the colored dots are the lidar points projected onto the image plane.

C. Depth Retrieval from Object Mask

In both configurations, we have a pixel-wise depth map containing either stereo or lidar depths available at each time frame. VLMOT uses lidar depth data, and VSMOT uses stereo depth data. Using the object mask for a tracked object and the depth map, we can create a vector $\mathbf{d}_{i,k}$ which contains all unique depth values of object i's mask at frame k. The vector $\mathbf{d}_{i,k}$ is sorted in ascending order and zero depths are removed. For figure 6b, this means collecting all the unique depth measures corresponding to the colored dots inside the object mask and sorting them. The object mask is not guaranteed to precisely cover the object. Thus, $\mathbf{d}_{i,k}$ may contain values from depth measurements behind the object. To counteract false depth values from behind the object, the median of $\mathbf{d}_{i,k}$ is chosen as the depth measurement for a tracked object.

D. Transformation from Camera Coordinates to World Coordinates

From BoT-SORT and YOLOv8's bounding boxes, we know the position of a tracked object in the image coordinate system. By using the depth measurement, we can find the position of a tracked object in the camera coordinate system. Let $\mathbf{x}_{i,k}^{\text{im}} = \begin{bmatrix} x & y \end{bmatrix}$ be the position of object i at frame k in the image coordinate system. Furthermore, let $\mathbf{x}_{i,k}^c = \begin{bmatrix} X & Y & Z & 1 \end{bmatrix}$ be the corresponding homogeneous point in the camera coordinate system, where Z is the known depth measurement. Then, the elements of $\mathbf{x}_{i,k}^c$ is given by using the pinhole camera model as follows

$$\begin{aligned} X &= Z\frac{x - x_0}{f_x} \\ Y &= Z\frac{y - y_0}{f_y} \end{aligned} \tag{2}$$

where x_0, y_0 is the coordinate of the principal point and f_x, f_y are the focal lengths in pixel units. To transform $\mathbf{x}_{i,k}^c$ to a fixed world coordinate system we need the camera pose at every time instant, which can be found through the pose of MA2. Using MA2's INS data we can, at every time instant, obtain a time-varying homogeneous transformation from MA2's body coordinate system $\{b\}$ to a fixed East-North-Up (ENU) coordinate system $\{e\}$. We let this transformation at time k be denoted as $\mathbf{H}_{b,k}^e$. This transformation also compensates for MA2's ego-motion and makes the pipeline robust against waves. Further, we let $\mathbf{H}_c^b$ be the homogeneous transformation matrix from the stereo camera to MA2's body frame. Then, $\mathbf{x}_{i,k}^c$ is transformed to a point in $\{e\}$ as

$$\mathbf{x}_{i,k}^e = \mathbf{H}_{b,k}^e \mathbf{H}_c^b \mathbf{x}_{i,k}^c. \tag{3}$$

E. Kalman Filtering of World Frame Estimates

Furthermore, we initialize a KF for each tracked object. The state vector is defined as $\mathbf{x} = \begin{bmatrix} x & y & \dot{x} & \dot{y} \end{bmatrix}^T$ in the $\{e\}$ coordinate system. The units for the position are in meters [m] and the velocities are in meters per second [$\mathrm{m\,s^{-1}}$]. We use a standard constant velocity model, as described in [29] p. 270. The discretization time T is initialized as $\frac{1}{15}$s as the video sequences are recorded at 15 Hz. Then, as soon as two measurements are available, the discretization time is time-varying and calculated as $T_k = t_k - t_{k-1}$, where t_k indicates the timestamp at frame k. This is necessary because the time between two frames may vary due to either frame drops or synchronization between the lidar measurements and the video. The $\mathbf{Q}$ matrix is given in [29]. The process noise standard deviation is set to $\sigma_Q = 0.5$. The $\mathbf{R}$ matrix is defined as $\mathbf{R} = \mathbf{I}_2 \sigma_R^2$ with $\sigma_R = 0.25$. The measurement matrix is given as $\mathbf{C} = \begin{bmatrix} \mathbf{I}_2 & \mathbf{0}_{2\times 2} \end{bmatrix}$, as we only measure the position. The KFs are initialized with the x, y components of the initial *a priori* estimate $\hat{\mathbf{x}}_0^-$ set to the initial position measurement

$\mathbf{x}_{i,0}^e$ and zero velocities. The initial *a priori* error covariance matrix $\mathbf{P}_0^-$ is initialized to the identity matrix $\mathbf{I}_4$.

F. Track Management

As mentioned, a track management module is implemented to keep track of lost tracks and match reappearing objects. Each time a new object is tracked, its ID, KF state vector and KF covariance matrix are stored in the track manager. Whenever an object is lost, i.e. not tracked by BoT-SORT anymore, its world frame position is predicted by its KF for 60 frames, and the state vector and covariance matrix are updated in the track manager. If BoT-SORT starts tracking a new object while a lost object's position is being predicted, the new object is attempted to be matched with any lost tracks. To obtain a match between a new object and a lost track, we calculate the Mahalanobis distance

$$D_M = \sqrt{(\mathbf{x}_{\text{lost},k} - \mathbf{x}_{\text{new},k})^T \mathbf{P}_{lost,k}^{-1} (\mathbf{x}_{\text{lost},k} - \mathbf{x}_{\text{new},k})} \quad (4)$$

where $\mathbf{x}_{\text{lost},k}$ is the predicted state of the lost track at frame k, $\mathbf{x}_{\text{new},k}$ is the state of the new track at frame k and $\mathbf{P}_{lost,k}$ is the covariance matrix of the KF belonging the lost track. If D_M is less than 10, the new object and the lost object are considered to be the same object. If no new object appears during the 60 frames the lost object's position is predicted, it is considered completely lost.

G. Mean Square Deviation

To provide a measure of the depth uncertainty for each modality, the mean square positive and negative deviation is calculated for each object mask. Recall that $\mathbf{d}_{i,k}$ is a vector consisting of all the unique and valid depths for object i at frame k, sorted in an ascending order. Further let $m_{i,k}$ be the median depth of this mask for object i at frame k. Then, we let $\mathbf{d}_{i,k}^+$ contain all the depths greater than $m_{i,k}$, and $\mathbf{d}_{i,k}^-$ contain all the depths smaller than $m_{i,k}$. Then, we calculate the mean square positive deviation for the $x\%$ percentile as

$$s_{x,k}^+ = \sqrt{\frac{1}{N_x^+} \sum_{j=1}^{N_x^+} (x_j - m_{i,k})^2} \quad (5)$$

where $N_x^+ = \text{round}(\frac{1}{x} N^+)$, where N^+ is the number of elements in $\mathbf{d}_{i,k}^+$. The mean square negative deviation can be calculated in the same manner, where $\mathbf{d}_{i,k}^-$ is sorted in descending order. Furthermore, the resulting depths $Z = m_{i,k}$, $Z = m_{i,k} - s_{x,k}^-$ and $Z = m_{i,k} + s_{x,k}^+$ are passed through the pipeline as explained above, resulting in three tracks for each object representing a uncertainty band. These results are shown in section V-D.

V. Results

A. SORT

To illustrate a challenge associated with inaccurate detections, we refer to figure 7. This ID-switch was primarily caused by a lack of detections for a few frames. Other reasons for ID-switches are variations in the size of the detection boxes due to

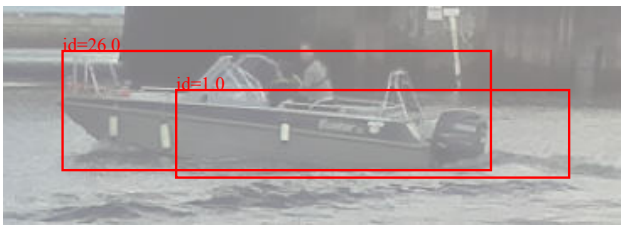

Fig. 7. Demonstration of an ID-switch: The target boat was undetected for several frames, leading to the SORT algorithm assigning a new ID to the track, changing it from 1 to 26.

background clutter, quick appearance or bounding box changes due to rotations in 3D, or big frame-to-frame movement due to frame drops in the video. There are numerous ways in which SORT can be improved, e.g., more advanced data association, improved detections, or using a 3D track manager as done in this paper. This observation supports the argument that the world motion provides a more robust cue for data association compared to the projected 2D motion in the image frame, as also discussed in the GNN3DMOT paper [30].

B. Boat Model

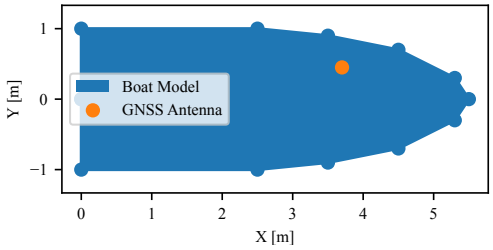

Fig. 8. Illustration of the boat model with physical measurements. The placement of the GNSS antenna inside the boat is marked in orange.

To provide a more accurate evaluation of the tracking results, tracks representing the shape of the boat are generated from the GNSS signals and considered the ground truth. As the measurements of the target Buster XL boat are known and the physical placement of the GNSS antenna, we created a boat model which is illustrated in figure 8. This shape model is used to generate the ground truth tracks visualized in figure 9. The creation of these tracks is explained in detail in [6].

C. Resulting Tracks

In figure 9a, the resulting tracks from VLMOT and VSMOT tested on episode 1 are shown. The estimated tracks are plotted together with the ground truth signals in a fixed world coordinate system. The target kayak, which is between 80 m to 35 m away from MA2, is only tracked for a few frames due to YOLOv8 struggling to detect it because it is too far away. Therefore, it is excluded from the plots. The boat is assigned ID 1. VLMOT tracks the boat accurately while VSMOT estimates the boat to be closer to MA2 than the ground truth signals.

Figure 9b shows the resulting tracks for VLMOT and VSMOT for episode 2. In this episode, the target kayak is

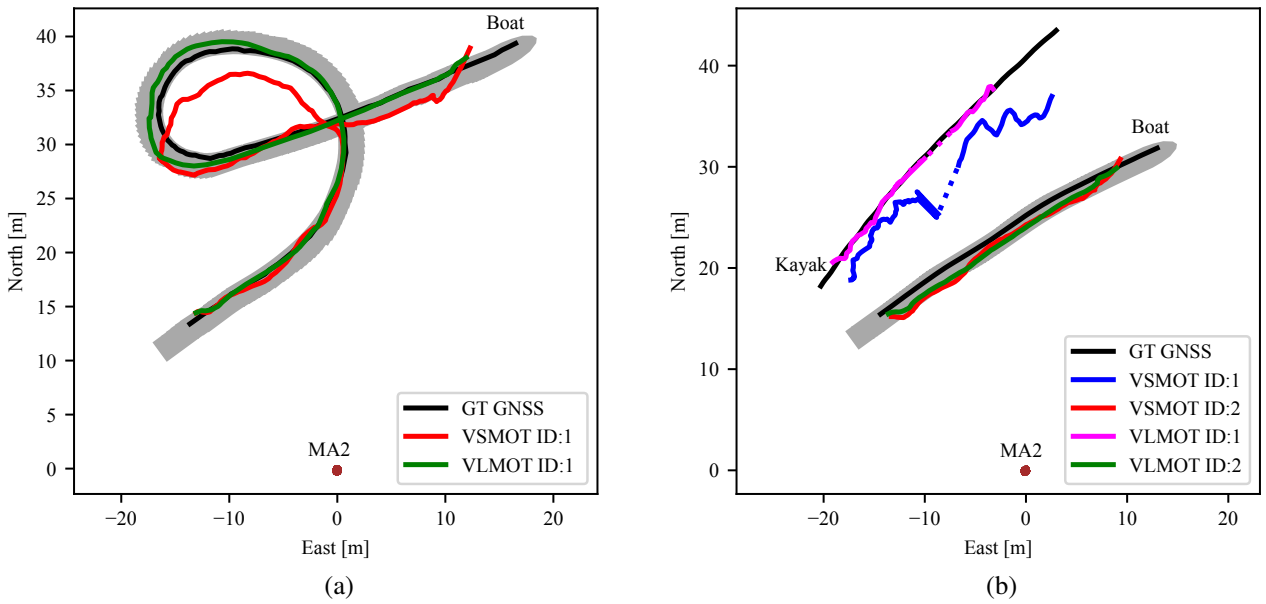

Fig. 9. Ground truth GNSS (GT GNSS), MA2 GNSS and resulting tracks for VLMOT and VSMOT tested on episode 1 (a) and episode 2 (b). The grey region is generated from the GNSS signal and represents how the shape of the boat moves during the episode. The dotted line segments represent the period when an object is occluded and its position is predicted. The ground truth signal for each target is cut to the starting and ending timestamp of the corresponding estimate.

assigned ID 1 and the boat ID 2. We see that the target boat is tracked accurately by both VLMOT and VSMOT. VLMOT is always inside the ground truth boat region. For the kayak, VLMOT tracks accurately, and predicts accurately while the kayak is occluded, indicated by the dotted line. For "VLMOT ID:1", the KF estimates move faster than the kayak actually does. Therefore, the predictions move past the point where the kayak is relocated, i.e. the solid line. VSMOT struggles with depth, leading to oscillatory tracks and prediction in the wrong direction during the occlusion. However, note that VSMOT starts tracking the kayak earlier than VLMOT as there are no lidar measurements available to begin with. The ground truth signal for the kayak is cut to fit the longest estimate, i.e. "VSMOT ID:1".

D. Root Mean Square Error (RMSE)

To provide a performance measure of the tracking configurations we use the Root Mean Square Error

$$\text{RMSE} = \sqrt{\frac{1}{n}\sum_{i=1}^{n}((x_i - \hat{x}_i)^2 + (y_i - \hat{y}_i)^2)}, \qquad (6)$$

where (x_i, y_i) is the ground truth position and $(\hat{x}_i, \hat{y}_i)$ is the corresponding positional estimate in the world coordinate system $\{e\}$. For each positional estimate, we find the ground truth point closest in time to calculate the squared error. For the target boat, we use the closest point between the estimate and the boat shape at the given timestamp. For the kayak, we employ the GNSS signal directly as the ground truth.

At the start and end of the VSMOT and VLMOT tracks, we see some irregular behavior in the estimates. This behavior occurs due to the object entering or exiting the image when the whole object is not visible in the image. As mentioned above, the tracking pipeline uses the center of the YOLOv8 bounding box as the point of measurement. Thus, as an object enters or exits the image, the point of measurement will be in the middle of what is visible of the given object, causing these artifacts. Therefore, for table I there is one value displayed for the whole sequence and one value without the start and ending, to account for the irregularities caused by the object's partial visibility. By excluding the initial and final frames from the analysis, we aim to provide a more accurate representation of the tracking performance.

TABLE I
RMSE FOR DIFFERENT CONFIGURATIONS FOR EPISODES 1 AND 2. "WO SE" MEANS WITHOUT START AND ENDING.

Episode	Configuration + ID	RMSE	RMSE wo se
1	VSMOT ID: 1	1.12	1.08
1	VLMOT ID: 1	**0.16**	**0.02**
2	VSMOT ID: 1	3.72	2.60
2	VLMOT ID: 1	**1.32**	1.34
2	VSMOT ID: 2	0.31	**0.06**
2	VLMOT ID: 2	**0.24**	**0.00**

Table I shows the RMSE values for VLMOT and VSMOT for both episodes. We see a clear benefit of using lidar as the depth modality, compared to using stereo. When not taking the start and ending of the estimates into account, VLMOT provides very accurate tracking.

E. Uncertainty Band

To provide a measure of uncertainty related to each depth modality, the mean square positive and negative deviations

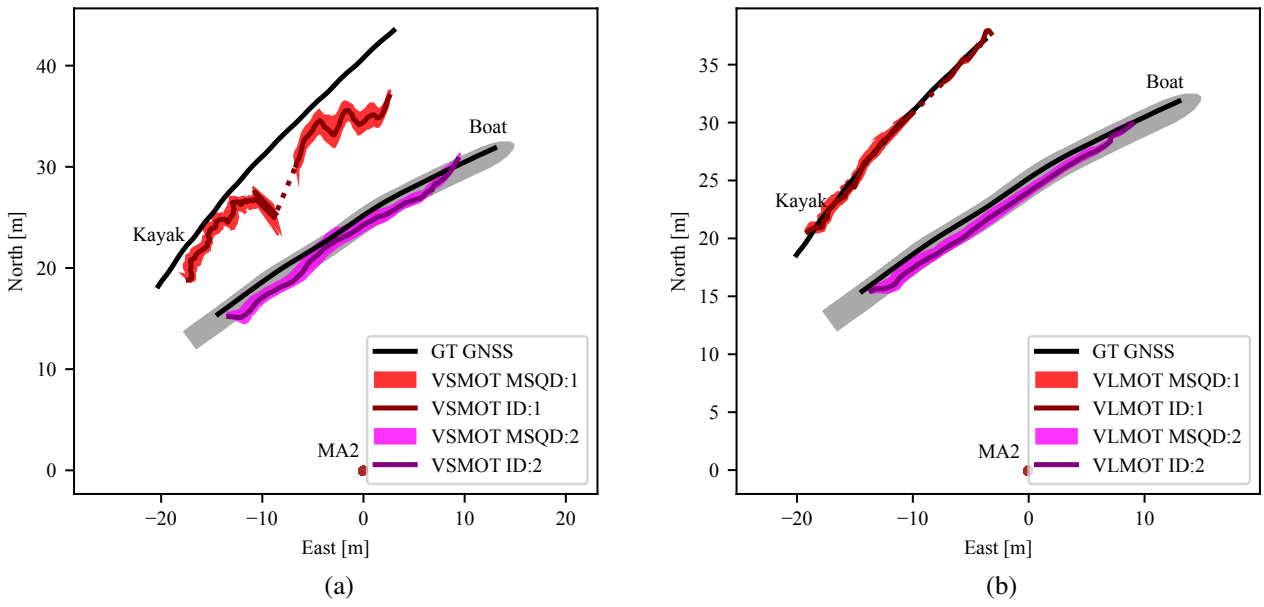

Fig. 10. Ground truth GNSS (GT GNSS), MA2 GNSS, resulting tracks and uncertainty band for VSMOT (a) and VLMOT (b) for episode 2. The mean square deviation (MSQD) represents the band created by the mean square positive and negative deviations. The dotted line segments represent the period when an object is occluded and its position is predicted.

are calculated as explained in section IV-G. The positive and negative deviations effectively create an uncertainty band for each tracked object, which represents the uncertainty for the object depths, which we have plotted. The $x\%$ percentile is set to 70% to remove noise from the depths behind the object. The resulting tracks as plotted in figure 9 are also plotted to show that the distribution is uneven, as the median, is set to $m_{i,k}$ in (5).

Figure 10 shows the uncertainty bands for VSMOT and VLMOT for episode 2. We see that both modalities are close to equally certain for the tracked boat with ID 2. However, there is a larger uncertainty for the tracked kayak, especially for VSMOT. Note that there is no depth-related uncertainty during the period when the kayak is occluded, as there are now depth measurements available.

F. Discussion

As stated earlier, the depth modality is the only thing that differs between the two configurations. Generally, VLMOT provides estimates closer to the ground truth than VSMOT. The VSMOT estimates have a larger error and fluctuate more for far-away objects. The ZED 1 camera has a baseline of $12\,\mathrm{cm}$, which is relatively short. For short baseline stereo cameras, the disparities of far-away points become very small, making it hard to accurately calculate the depth. In episode 2, the kayak is at most about $46\,\mathrm{m}$ away and the VSMOT estimates fluctuate a lot, as seen in figure 9b. However, the boat is relatively close and the VSMOT and VLMOT estimates coincide. To summarize, the stereo camera struggles to measure depth when objects are far away. Essentially, there is a trade-off between the depth accuracy and the baseline. As mentioned, the depth map used is retrieved directly from ZED's software. Thus, the VSMOT results rely on the quality of ZED's depth calculations.

As mentioned, when the boat enters or leaves the camera view, we have some artifacts in the plots. There is a change of direction in the positional estimates and the GNSS signal extends beyond the estimate. This is because when an object enters or leaves the image, the bounding box only covers part of the object. When the object leaves the image, this phenomenon causes a change of direction for the point of measurement generated by the pipeline. The GNSS signal extends beyond the estimate when the object exits the image because the pipeline uses a point on the back of the object as its point of measurement, causing a distance between the estimate and ground truth.

Another interesting observation is that both configurations have fluctuations, but VSMOT more so than VLMOT due to using stereo depths. Part of the fluctuations present in both configurations most likely originate from the fluctuating size of bounding boxes. This fluctuation causes the center of the bounding box to move, which is the point that is used further in the pipeline. In addition, the median depth may jump from frame to frame, causing a fluctuating depth measurement. Hence, these fluctuations propagate through all transformations and to the world coordinates. The KF somewhat counteracts this with its smoothing effect.

The constant velocity model is used to model the movement of the tracked target, which assumes that the object is moving in a specific direction at a constant speed. This is the situation for episode 2, but not for episode 1. If an object is occluded while turning, the KF predictions will not take into account the turning rate and predict erroneously.

A drawback of the tracking pipeline is the dependence on

a good object detector. The pipeline is limited by YOLOv8's ability to detect small objects in the image. For episode 1, there is a kayak in the background. The depth from MA2 to the kayak ranges from $80\,\mathrm{m}$ to $35\,\mathrm{m}$ during the sequence, which causes the kayak to appear very small in the image. YOLOv8 struggles to detect the kayak, causing the whole pipeline to fail. Also, the pipeline is not able to track all kinds of objects as YOLOv8 is trained on 80 classes. Thus, it will not be able to detect objects like a floating log.

VI. Conclusion

In this paper, a tracking pipeline for autonomous surface vehicles is proposed. The pipeline utilizes a camera for object detection and tracking, and a stereo camera or lidar sensor for depth measurements, resulting in two configurations; VLMOT and VSMOT. The pipeline also incorporates track management that handles occlusions. VLMOT and VSMOT are tested on two different episodes from a real dataset involving NTNU's MA2 ferry. The results from these experiments are further analyzed and discussed, and the two configurations are compared to each other. In general, VLMOT is more accurate and less noisy for far-away objects. As the pipeline's biggest drawback is the range limitation caused by YOLOv8, not the lidar sensor, a natural direction for future work would be to fuse lidar detections into the pipeline and make 3D associations.

References

[1] T. I. Fossen, *Handbook of marine craft hydrodynamics and motion control*. John Wiley & Sons, 2021.

[2] Ø. K. Helgesen, E. H. Thyri, E. F. Brekke, A. Stahl, and M. Breivik, "Experimental validation of camera-based maritime collision avoidance for autonomous urban passenger ferries," *Modeling, Identification and Control*, vol. 44, no. 2, pp. 55–68, 2023.

[3] M. T. Wolf, C. Assad, Y. Kuwata, A. Howard, H. Aghazarian, D. Zhu, T. Lu, A. Trebi-Ollennu, and T. Huntsberger, "360-Degree Visual Detection and Target Tracking on an Autonomous Surface Vehicle," *Journal of Field Robotics*, vol. 27, no. 6, pp. 819–833, 2010.

[4] 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.

[5] A. Bewley, Z. Ge, L. Ott, F. Ramos, and B. Upcroft, "Simple online and realtime tracking," in *2016 IEEE International Conference on Image Processing (ICIP)*, pp. 3464–3468, IEEE, 2016.

[6] N. Dalhaug, A. Stahl, R. Mester, and E. F. Brekke, "Combining short and wide baseline stereo cameras for improved maritime target tracking." Submitted to FUSION2024.

[7] W. Luo, J. Xing, A. Milan, X. Zhang, W. Liu, and T.-K. Kim, "Multiple object tracking: A literature review," *Artificial Intelligence*, vol. 293, p. 103448, 2021.

[8] N. Wojke, A. Bewley, and D. Paulus, "Simple online and realtime tracking with a deep association metric," in *2017 IEEE International Conference on Image Processing (ICIP)*, pp. 3645–3649, IEEE, 2017.

[9] Y. Du, Z. Zhao, Y. Song, Y. Zhao, F. Su, T. Gong, and H. Meng, "StrongSORT: Make DeepSORT Great Again," *IEEE Transactions on Multimedia*, vol. 25, pp. 8725–8737, 2023.

[10] J. Cao, J. Pang, X. Weng, R. Khirodkar, and K. Kitani, "Observation-Centric SORT: Rethinking SORT for Robust Multi-Object Tracking," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 9686–9696, 2023.

[11] G. Maggiolino, A. Ahmad, J. Cao, and K. Kitani, "Deep OC-SORT: Multi-Pedestrian Tracking by Adaptive Re-Identification," *arXiv preprint arXiv:2302.11813*, 2023.

[12] Y. Zhang, P. Sun, Y. Jiang, D. Yu, F. Weng, Z. Yuan, P. Luo, W. Liu, and X. Wang, "ByteTrack: Multi-Object Tracking by Associating Every Detection Box," in *European Conference on Computer Vision*, pp. 1–21, Springer, 2022.

[13] N. Aharon, R. Orfaig, and B.-Z. Bobrovsky, "BoT-SORT: Robust Associations Multi-Pedestrian Tracking," *arXiv preprint arXiv:2206.14651*, 2022.

[14] A. Kim, A. Ošep, and L. Leal-Taixé, "EagerMOT: 3D Multi-Object Tracking via Sensor Fusion," in *2021 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 11315–11321, IEEE, 2021.

[15] A. Geiger, P. Lenz, and R. Urtasun, "Are We Ready for Autonomous Driving? the KITTI Vision Benchmark Suite," in *2012 IEEE Conference on Computer Vision and Pattern Recognition*, pp. 3354–3361, IEEE, 2012.

[16] H. Wu, W. Han, C. Wen, X. Li, and C. Wang, "3D Multi-Object Tracking in Point Clouds Based on Prediction Confidence-Guided Data Association," *IEEE Transactions on Intelligent Transportation Systems*, vol. 23, no. 6, pp. 5668–5677, 2022.

[17] Y. Li, J. Deng, Y. Zhang, J. Ji, H. Li, and Y. Zhang, "ezfusion: A close look at the integration of lidar, millimeter-wave radar, and camera for accurate 3d object detection and tracking," *IEEE Robotics and Automation Letters*, vol. 7, no. 4, pp. 11182–11189, 2022.

[18] H. Caesar, V. Bankiti, A. H. Lang, S. Vora, V. E. Liong, Q. Xu, A. Krishnan, Y. Pan, G. Baldan, and O. Beijbom, "nuScenes: A multimodal dataset for autonomous driving," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 11621–11631, 2020.

[19] L. Elkins, D. Sellers, and W. R. Monach, "The Autonomous Maritime Navigation (AMN) project: Field tests, autonomous and cooperative behaviors, data fusion, sensors, and vehicles," *Journal of Field Robotics*, vol. 27, no. 6, pp. 790–818, 2010.

[20] T. Huntsberger, H. Aghazarian, A. Howard, and D. C. Trotz, "Stereo vision–based navigation for autonomous surface vessels," *Journal of Field Robotics*, vol. 28, no. 1, pp. 3–18, 2011.

[21] J. Muhovič, B. Bovcon, M. Kristan, J. Perš, *et al.*, "Obstacle tracking for unmanned surface vessels using 3-D point cloud," *IEEE Journal of Oceanic Engineering*, vol. 45, no. 3, pp. 786–798, 2019.

[22] M. Sorial, I. Mouawad, E. Simetti, F. Odone, and G. Casalino, "Towards a Real Time Obstacle Detection System for Unmanned Surface Vehicles," in *OCEANS 2019 MTS/IEEE SEATTLE*, (Seattle, WA, USA), pp. 1–8, IEEE, Oct. 2019.

[23] 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.

[24] Z. Yao, X. Chen, and C. Shi, "Research on surface environment perception via camera-lidar sensor fusion," in *2023 6th International Conference on Artificial Intelligence and Big Data (ICAIBD)*, pp. 895–899, 2023.

[25] T. Clunie, M. DeFilippo, M. Sacarny, and P. Robinette, "Development of a perception system for an autonomous surface vehicle using monocular camera, lidar, and marine radar," in *2021 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 14112–14119, 2021.

[26] E. F. Brekke, E. Eide, B.-O. H. Eriksen, E. F. Wilthil, M. Breivik, E. Skjellaug, O. K. Helgesen, A. M. Lekkas, A. B. Martinsen, E. H. Thyri, T. Torben, E. Veitch, O. A. Alsos, and T. A. Johansen, "milliAmpere: An Autonomous Ferry Prototype," *Journal of Physics: Conference Series*, vol. 2311, p. 012029, July 2022.

[27] G. Jocher, A. Chaurasia, and J. Qiu, "Ultralytics YOLO," 2023.

[28] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick, "Microsoft COCO: Common objects in context," in *Computer Vision–ECCV 2014: 13th European Conference, Zurich, Switzerland, September 6-12, 2014, Proceedings, Part V 13*, pp. 740–755, Springer, 2014.

[29] Y. Bar-Shalom, X. R. Li, and T. Kirubarajan, *Estimation with Application to Tracking and Navigation*. Wiley, 2001.

[30] X. Weng, Y. Wang, Y. Man, and K. M. Kitani, "Gnn3dmot: Graph neural network for 3d multi-object tracking with 2d-3d multi-feature learning," in *CVPR*, June 2020.