IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received 28 February 2023, accepted 17 March 2023, date of publication 24 March 2023, date of current version 29 March 2023.

Digital Object Identifier 10.1109/ACCESS.2023.3261556

RESEARCH ARTICLE

Maritime Tracking With Georeferenced Multi-Camera Fusion

ØYSTEIN K. HELGESEN, ANNETTE STAHL, AND EDMUND F. BREKKE, (Senior Member, IEEE)
Department of Engineering Cybernetics, Norwegian University of Science and Technology (NTNU), 7034 Trondheim, Norway

Corresponding author: Øystein K. Helgesen (oystein.k.helgesen@ntnu.no)

This work was supported in part by the Research Council of Norway under Project 223254, Project 295033, and Project 309230; and in part by Norwegian University of Science and Technology (NTNU) through the Autoferry Project.

ABSTRACT Cameras form an essential part of any autonomous surface vehicle's sensor package, both for COLREGs compliance to detect light signals and for identifying and tracking other vessels. Due to limited fields of view compared to more traditional autonomy sensors such as lidars and radars, an autonomous surface vessel will typically be equipped with multiple cameras which can induce biases when used in tracking if a target is present in multiple image frames. In this work, we propose a novel pipeline for camera-based maritime tracking that combines georeferencing with clustering-based multi-camera fusion for bias-free camera measurements with target range estimates. Using real-world datasets collected using the *milliAmpere* research platform the performance of this pipeline exceeded a lidar benchmark across multiple performance measures, both in pure detection performance and as part of a JIPDA-based tracking system.

INDEX TERMS Sensor fusion, target tracking, situational awareness, autonomous surface vessel, experimental validation.

I. INTRODUCTION

Autonomous surface vehicles (ASVs) are often equipped with multiple heterogeneous sensors both for redundancy and robustness purposes. Sensors with differing modalities and operating principles can have complementary strengths, increasing both the performance and the reliability of the vehicle's situational awareness systems. One such example is how at shorter ranges a lidar, which provides very accurate, high-frequency sensor data, complements a radar that has a far greater range but at a lower frequency and with less accurate sensor data [1]. If one of the sensors fails the vehicle will also be able to maintain a reduced level of situational awareness that might be enough for safe navigation.

In recent years imaging sensors such as daylight electro-optical (EO) and infrared (IR) cameras have seen increased use in autonomous systems. Compared to the commonly used active sensors cameras can yield higher update frequencies and greater resolution allowing the situational awareness system to both detect and classify targets of interest within sensor range. The automotive sector was an early adopter of imaging sensors, finding applications for both driver assistance systems [2] and autonomy [3]. Maritime applications typically combine cameras with other sensors such as radar [4], [5] or AIS [6], however, some applications of camera-only tracking exist [7], [8].

Cameras are not without their challenges. Due to the lack of active signal emission imaging data only encodes the direction of the light that is gathered at a pixel and not the distance to the light source. For sensor redundancy purposes this presents some unique challenges when only passive sensors remain functional. While several methods for bearings-only tracking have been developed [9], [10], [11], they all require the vehicle to perform a series of significant maneuvers to induce observability. In many cases, this is not desirable behavior due to energy efficiency, passenger comfort, etc. Camera-based depth estimation has therefore seen significant research focus in recent years. The advent of deep learning has seen the introduction of several methods for monocular depth estimation based on neural networks [12], [13] with promising performance. However, these methods rely on large datasets and can be computationally expensive, often failing when encountering data significantly different from their training set.

The associate editor coordinating the review of this manuscript and approving it for publication was Junho Hong.

Binocular depth estimation, also known as stereo vision, is an alternate approach that relies on triangulating two cameras with overlapping fields of view. Pixels in each image are matched to each other using a wide range of methods from classical [14] to deep learning-based [15]. These methods are less reliant on large training data but can still be computationally expensive and large baselines are typically required for most distances encountered in a maritime context. They also increase the cost and complexity of the sensor suite, requiring twice the number of cameras for equivalent coverage. A third, hybrid approach is structure-from-motion [16] which emulates a stereo vision setup through the motion of a monocular camera, yielding multiple views of an object from a single camera. This removes the need for binocular cameras, but can still require significant computation. Non-stationary objects also pose an issue due to the time difference between the matched images.

In contrast, active sensors such as radar and lidar will return measurements of both the target range and bearing without the need for complex processing or estimation. Cameras are also typically equipped with lenses yielding limited fields of view, requiring multiple cameras to provide sensor coverage in all directions. In certain positions, a target might therefore be partially present in the sensor frames of two or more cameras. If the cameras are processed individually by the tracking system, this will induce biases in the state estimates as only parts of the vessel would be visible in any one frame. If processed collectively as a single, virtual sensor, more than one measurement would originate from the target in question violating a common assumption in many tracking methods [17], [18], [19].

In this work, we propose a multi-camera detection pipeline that circumvents these issues. Adapting the georeferencing method presented in [20] to a moving platform allows the usage of implicit information to estimate target ranges from image detections, eliminating the need for bearings-only tracking or complex monocular or stereo-based depth estimation. Clustering-based sensor fusion is then used to perform measurement-level sensor fusion of the different cameras, yielding only a single measurement from each target. This approach also allows for multi-spectral, pre-tracking fusion of IR and EO cameras. A real-world dataset is used to evaluate the detection performance of this pipeline against a lidar benchmark using two distinct targets equipped with accurate GNSS ground truth sensors. We also integrate this pipeline into a multi-target, multi-sensor tracking system based on Joint Integrated Probabilistic Data Association (JIPDA) [21] and evaluate it against the lidar benchmark.

II. SENSOR PLATFORM

The research platform milliampere shown in Fig. 1, is an urban autonomous passenger ferry developed by the Autoferry project at NTNU. The sensor system of the ferry is described in detail in [1] and [22], this section repeats key details used in this work.

FIGURE 1. ***Milliampere 1*** **(background) and** ***milliAmpere 2*** **(foreground).**

To maximize research potential *milliAmpere* is equipped with multiple exteroceptive and proprioceptive sensors. Navigation is provided by a dual antenna GNSS navigation system with real-time kinematic corrections from a land-based station. *milliAmpere* is also equipped with multiple situational awareness sensors where some have been used in this work. The lidar benchmark utilizes a Velodyne VLP-16 sensor with a maximum range of 100m. Imaging data is provided by a 360° camera rig equipped with 5 EO and 5 IR cameras. These cameras operate at 5 and 9 HZ with resolutions of 1224×1024 and 640×512 pixels.

III. IMAGE PROCESSING

Machine vision cameras typically equipped on ASVs do not always yield sensor data readily processable by common computer vision algorithms. In this section, we introduce the pre-processing steps applied to *milliAmpere's* camera data before detection pipeline processing.

A. COLOR CONVERSION

Most commercially available RGB daylight cameras utilize Bayer-type imaging sensors first described in a 1976 patent [23] by B. Bayer. These sensors consist of a grid of photosensors corresponding to individual pixels where each sensor only captures a single color, either red, green or blue. Bayer aligned the photosensors in an alternating grid with half green and a quarter red and blue sensors shown in Fig. 2. This pattern was based on how the human eye perceives light, more specifically the luminance, or brightness, and the color, chrominance. Human eyes have greater sensitivity to luminance than chrominance and the M and L cones responsible for this are more responsive to green than the other colors. Mimicking this sensitivity results in higher luminance resolution and was expected to yield a better-looking image compared to the equal color allocation which has greater chrominance resolution. The resulting sensor output is known as a Bayer pattern image, which

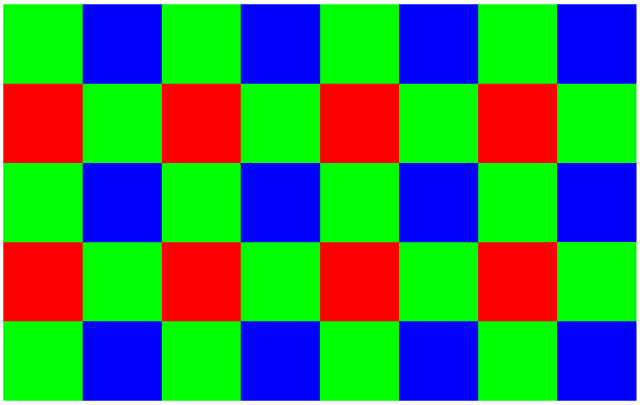


FIGURE 2. Bayer photosensor color alignment.

is, on a pixel level, monochrome. An example is shown in Fig. 3a.

To convert the Bayer pattern image to a full-color image a process known as demosaicing must be applied. Many algorithms exist for this purpose ranging from simple interpolation to more complex correlation-based methods [24]. Many machine vision cameras are capable of outputting either raw, Bayer pattern images or demosaiced RGB color images, shown in Fig. 3b. The latter shifts the responsibility for implementing demosaicing from the user to the camera and could also be more computationally efficient and faster due to application-specific digital signal processors in the camera. The former allows the user to choose between a wider range of demosaicing methods and also reduces the bandwidth required to 1/3rd as the image can be transmitted as monochrome, requiring only a single intensity for each pixel. This is also the main motivation for outputting Bayer format images from *milliAmpere's* sensor rig which is connected by a single 1Gb ethernet link with limited bandwidth.

B. DISTORTION CORRECTION

Most camera lenses yield a non-perfect projection of light onto the image sensor, resulting in a deviation from the expected versus the actual pixel a ray intersects with. This phenomenon is known as lens distortion and primarily appears in two forms, radial and tangential. Radial distortion is approximately symmetric and causes straight lines to appear curved. Tangential distortion appears when lens elements are not perfectly aligned with the sensor plane, resulting in some elements appearing closer than expected. Radial distortion is typically modeled using two or three distortion coefficients [25] according to

$$x_d = x + k_1 r^2 + k_2 r^4 + k_3 r^6 \tag{1}$$

$$y_d = y + k_1 r^2 + k_2 r^4 + k_3 r^6 \tag{2}$$

$$r^2 = (x^2 + y^2) \tag{3}$$

where x_d and y_d are the distorted pixel coordinates of the expected coordinates x and y while k_{1-3} are the radial distortion coefficients. Tangential distortion is typically modeled

(a) Raw Bayer pattern image.

(b) Demosaiced image.

(c) Demosaiced, undistorted image.

FIGURE 3. Pre-processing pipeline example of image captured in the Trondheimsfjord from *milliAmpere*.

using two coefficients according to

$$x_d = x + 2p_1xy + p_2(r^2 + 2x^2) \tag{4}$$
$$y_d = y + p_1(r^2 + 2y^2) + 2\,p_2xy) \tag{5}$$

where p_{1-2} are the tangential distortion coefficients. Combining these yield

$$x_d = x + k_1r^2 + k_2r^4 + k_3r^6 + 2p_1xy + p_2(r^2 + 2x^2) \tag{6}$$
$$y_d = y + k_1r^2 + k_2r^4 + k_3r^6 + 2\,p_2xy + p_1(r^2 + 2y^2). \tag{7}$$

Many open-source frameworks, including the Robot Operating System and OpenCV, estimate these parameters as part of their camera calibration process and also include functionality for correcting lens distortion. The end result of the pre-processing steps is an RGB color image corrected for lens distortion, shown in Fig. 3c.

IV. CAMERA DETECTION PIPELINE

For the raw imaging data to be useful in a target tracking system, multiple processing steps must be performed, forming a detection pipeline. In this section, we detail the various parts of this pipeline used to generate sensor measurements from this data.

A. DETECTOR

Sensor data from cameras is usually supplied in the form of color images with three 8-bit channels per pixel, each representing either red, green, or blue intensity corresponding to the wavelength of light that hit that pixel. In contrast to active sensors such as radar and lidar which typically do not yield substantial returns from the ocean surface, a camera will yield color information for all parts of a scene. Interpreting this sensor data thus requires more complex detectors capable of separating objects of interest from background noise.

Research interest in computer vision has increased rapidly in the last decade. The commoditization of computing power has made it possible to train highly accurate deep-learning-based models with millions of parameters for the detection and classification of objects in images. In this work we utilize a detector based on the Yolo v4 architecture [26] trained on the COCO dataset [27], however, the camera pipeline described in this work is detector agnostic and any other detector yielding a bounding box or segmented outline will work. An example detection output is shown in Fig. 4.

B. RANGE ESTIMATION

Due to their passive nature, cameras do not natively supply pixel ranges requiring further processing for range information to be extracted. The key element in this process is to utilize implicit information to estimate target ranges based on pixel detections. For maritime path planning and collision avoidance, only a certain subset of objects such as boats and kayaks need actual tracking. These objects will almost always be situated on the ocean surface, which in calmer conditions can be modeled fairly accurately as a flat plane.

FIGURE 4. Yolo v4 detection output.

By placing the camera above this plane, the target position can be estimated using triangulation without the need for more complex stereo camera systems.

This process requires an accurate measurement of the camera's extrinsic parameters, i.e. its position and orientation relative to a local NED coordinate frame. This information is typically supplied by a vessel's navigation where sensor data from IMUs and GNSS receivers provide the vessel's pose and position. A simple, fixed transform from the vessel body frame will then yield the extrinsic camera parameters. This transform can easily be found using a 3D model of the vessel, physical measurements, or estimated based on the camera's intrinsic parameters and known object positions. We denote this transform as a combined translation vector, $\mathbf{t}_c^w$, and a rotation matrix, $\mathbf{R}_c^w$, from the camera frame c to the local NED frame, w, given by

$$\mathbf{R}_c^w = \mathbf{R}_b^w(t)\mathbf{R}_c^b \tag{8}$$
$$\mathbf{t}_c^w = \mathbf{R}_b^w(t)\mathbf{t}_c^b + \mathbf{t}_b^w(t) \tag{9}$$

where $\mathbf{R}_b^w(t)$ and $\mathbf{t}_b^w(t)$ is the dynamic transform from body frame b to the NED frame w at time t supplied by the navigation system.

The camera's intrinsic parameters according to the pinhole camera model include the focal length of the lens, f_x and f_y, as well as the optical center c_x and c_y. Combined with the extrinsic parameters this is all that is necessary to project the world point $\mathbf{x}_w = \begin{bmatrix} x_w & y_w & z_w \end{bmatrix}^\mathsf{T}$ into pixel coordinates $\mathbf{x}_p = \begin{bmatrix} x_p & y_p \end{bmatrix}^\mathsf{T}$:

$$s\begin{bmatrix}\mathbf{x}_p \\ 1\end{bmatrix} = \mathbf{K}[\mathbf{R}_w^c|\mathbf{t}_w^c]\begin{bmatrix}\mathbf{x}_w \\ 1\end{bmatrix} \tag{10}$$

where s is a scale factor given by the depth of the point in the camera frame and $\mathbf{K}$ is the intrinsic matrix given by

$$\mathbf{K} = \begin{bmatrix} f_x & 0 & c_x \\ 0 & f_y & c_y \\ 0 & 0 & 1 \end{bmatrix}. \tag{11}$$

Inversely, if one or more of the target origin coordinates are known the pinhole model can be reversed to recover the origin point of a pixel. Denoting the combined camera matrix $\mathbf{P}$ as

$$\mathbf{P} = \mathbf{K}[\mathbf{R}_w^c|\mathbf{t}_w^c] = \begin{bmatrix} p_{11} & p_{12} & p_{13} & p_{14} \\ p_{21} & p_{22} & p_{23} & p_{24} \\ p_{31} & p_{32} & p_{33} & p_{34} \end{bmatrix} \tag{12}$$

the pinhole projection model of (10) can be rearranged to yield

$$\begin{bmatrix} x_p p_{31} - p_{11} & x_p p_{32} - p_{12} \\ y_p p_{31} - p_{21} & y_p p_{32} - p_{22} \end{bmatrix} \begin{bmatrix} x_w \\ y_w \end{bmatrix} = \begin{bmatrix} z_w(p_{13} - x_p p_{33}) + p_{14} - x_p p_{34} \\ z_w(p_{23} - y_p p_{33}) + p_{24} - y_p p_{34} \end{bmatrix}. \tag{13}$$

For distances within the camera detection range the earth curvature induced elevation change between the ownship and the detected target is negligible, $\Delta z_w < 0.02$m for distances up to 500m, and the pixel origin elevation can therefore be assumed to be identical to the ownship elevation. This elevation can be supplied by the navigation system of the ownship, either as a time-moving average to compensate for measurement noise and heave motion or as the current instantaneous estimate. This results in

$$\begin{bmatrix} x_w \\ y_w \end{bmatrix} = \begin{bmatrix} x_p p_{31} - p_{11} & x_p p_{32} - p_{12} \\ y_p p_{31} - p_{21} & y_p p_{32} - p_{22} \end{bmatrix}^{-1} \times \begin{bmatrix} z_w(p_{13} - x_p p_{33}) + p_{14} - x_p p_{34} \\ z_w(p_{23} - y_p p_{33}) + p_{24} - y_p p_{34} \end{bmatrix} \tag{14}$$

which yields the position of the pixel origin.

C. DETECTION AGGREGATION

For each bounding box outputted by the detector, position estimates are calculated for the pixel positions $\begin{bmatrix} x_{min} & y_{max} \end{bmatrix}$ and $\begin{bmatrix} x_{max} & y_{max} \end{bmatrix}$ corresponding to the bottom corners of the bounding box. To maintain the consistency of the clustering-based camera fusion described in section IV-E, additional position estimates are generated between the two corners using linear interpolation with a range of maximum 1m between subsequent estimates as shown in Fig. 5. This distance was selected based on expected target separation. Vessels are unlikely to operate at distances closer than this which should avoid merging detections from distinct targets. Denoting the position estimate of the left corner as $\mathbf{x}_w^0 = \begin{bmatrix} x_w^0 & y_w^0 \end{bmatrix}^\top$ and the right corner as $\mathbf{x}_w^1 = \begin{bmatrix} x_w^1 & y_w^1 \end{bmatrix}^\top$, the range r between the two estimates is given by

$$r = \sqrt{(x_w^1 - x_w^0)^2 + (y_w^1 - y_w^0)}. \tag{15}$$

The total number of points needed to ensure the distance threshold is maintained along the width of the bounding box is then

$$N_w = \left\lceil \frac{r}{T_d} \right\rceil \tag{16}$$

where T_d is the specified threshold and $\lceil y \rceil$ the ceiling function of y. The difference in x and y between subsequent estimates is given by

$$\Delta_x = \frac{(x_w^1 - x_w^0)}{N} \tag{17}$$

$$\Delta_y = \frac{(y_w^1 - y_w^0)}{N} \tag{18}$$

which yields the calculation

$$\mathbf{x}_w^i = \left\{ \mathbf{x}_w^0 + i \begin{bmatrix} \Delta_x \\ \Delta_y \\ 0 \end{bmatrix} \mid i \in \mathbb{Z}, 0 < i < N_w \right\} \tag{19}$$

for estimate i. This process, including detection and range estimation, is repeated for all cameras present in the system and the estimates, including interpolations, are aggregated into a single point cloud in the local NED frame.

1) TARGET HEIGHT ESTIMATION

Target height is also estimable using a similar approach. Assuming all pixels are equidistant we can find the length of a single pixel, l_p, using the calculation

$$l_p = \frac{\|\mathbf{x}_w^N - \mathbf{x}_w^0\|}{x_{max} - x_{min}}. \tag{20}$$

While there will be some minor variations in pixel size as the corners of the bounding box are further away than the center, this assumption should be reasonably accurate within the camera's detection range. Once the average pixel size is known the target height is estimated according to

$$h_t = l_p * (y_{max} - y_{min}). \tag{21}$$

The total number of vertical points required to ensure the specified distance threshold is met is similarly calculated according to

$$N_h = \lceil \frac{h_t}{T_d} \rceil. \tag{22}$$

Denoting $\mathbf{x}_w^2$ as the position estimate of the top right corner and $\mathbf{x}_w^3$ as the top left corner of the bounding box, calculated according to

$$\mathbf{x}_w^2 = \mathbf{x}_w^1 + \begin{bmatrix} 0 \\ 0 \\ t_h \end{bmatrix} \tag{23}$$

$$\mathbf{x}_w^3 = \mathbf{x}_w^0 + \begin{bmatrix} 0 \\ 0 \\ t_h \end{bmatrix}, \tag{24}$$

the points along the right side of the bounding box are generated according to

$$\mathbf{x}_w^i = \left\{ \mathbf{x}_w^1 + i \begin{bmatrix} 0 \\ 0 \\ \Delta_z \end{bmatrix} \mid i \in \mathbb{Z}, 0 < i < N_h \right\} \tag{25}$$

where

$$\Delta_z = h_t / N. \tag{26}$$

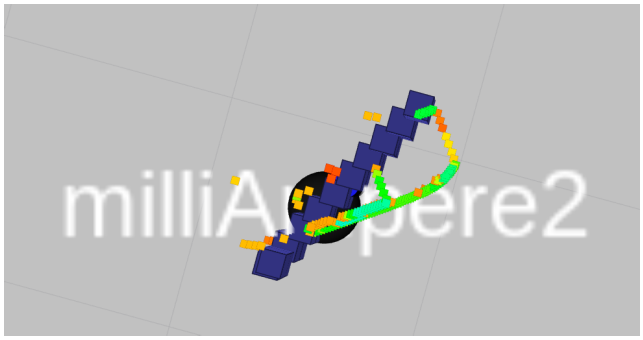


FIGURE 5. Bounding box extent estimation, visualized as blue squares, and lidar point cloud, colored dots.

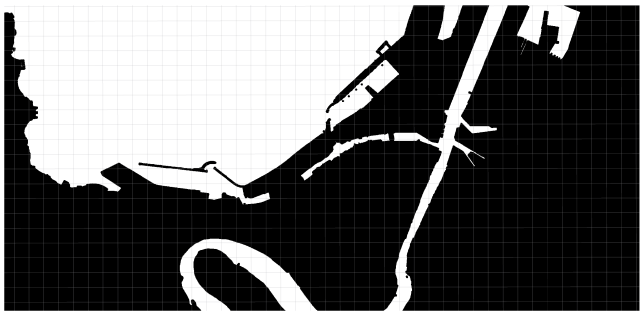


FIGURE 6. Occupancy grid of Trondheim area. White areas signify water, black areas land.

Points along the top of the bounding box can be found according to

$$\mathbf{x}_w^i = \left\{ \mathbf{x}_w^3 + i \begin{bmatrix} \Delta_x \\ \Delta_y \\ 0 \end{bmatrix} \mid i \in \mathbb{Z}, 0 < i < N_w \right\} \tag{27}$$

and points along the left side of the bounding box according to

$$\mathbf{x}_w^i = \left\{ \mathbf{x}_w^0 + i \begin{bmatrix} 0 \\ 0 \\ \Delta_z \end{bmatrix} \mid i \in \mathbb{Z}, 0 < i < N_h \right\} \tag{28}$$

D. DETECTION FILTERING

Littoral environments often contain moorings with multiple stationary vessels. While technically still valid targets, tracking these vessels might put an unneeded computational strain on the autonomy system with the possibility of real-time performance degradation. This could drastically impact the safety of the ownship, especially when maneuvering close to other vessels. Moored vessels are also unlikely to move, and, if tracked, more computationally efficient data association methods might suffice. For these reasons, the possibility to label or even filter out detections from these vessels is of high usefulness in a tracking system.

1) OCCUPANCY GRID

Occupancy grids are a family of map-generating algorithms based on noisy sensor data. Environments are represented as a binary grid where each cell has a binary value corresponding to either occupied, i.e. obstacle, or unoccupied. By generating a local map, either through SLAM or other methods, where unoccupied cells represent valid target positions, any detections falling outside these cells can be labeled or filtered out.

2) MAP GENERATION

Based on data from the Norwegian Mapping Authority a base map is generated fixed in a local NED frame. This base map covers all land areas and is dilated slightly to compensate for navigation and sensor uncertainty. Jetties and other potential mooring structures often extend further into the water than what the dilated base map covers, requiring additional masking. This is done manually using freely available online tools but could also be performed automatically using lidar or visual SLAM to generate the additional masks. The end result of this process is shown in Fig. 6.

E. CAMERA FUSION

In autonomous platforms with adjacent or overlapping camera fields of view, situations might arise where a target is visible for multiple cameras. If processed individually, the resulting detection output could contain significant biases if only parts of the vessel are visible. If processed collectively as a single virtual sensor, this might also result in multiple measurements per target. Both of these issues can be mitigated by fusing detections using clustering.

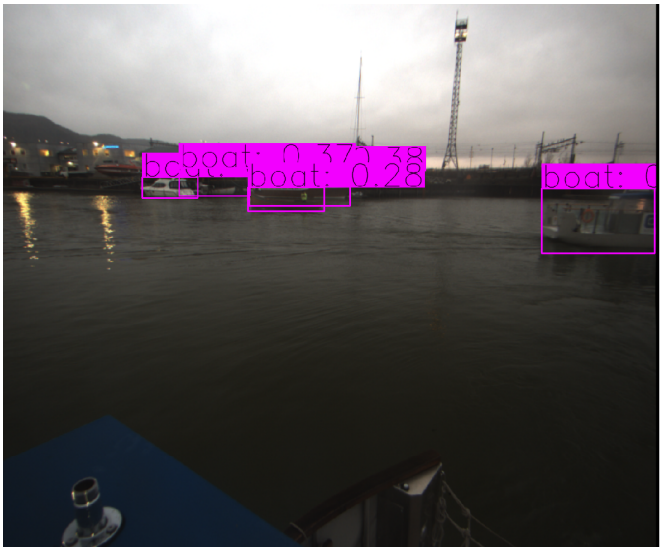
The clustering algorithm used in this pipeline is based on single link hierarchical clustering with k-d tree optimizations and was first described in [28] for clustering radar data. Each individual point gives rise to a cluster. Clusters are then merged if any individual points in the two clusters i and j, $\mathbf{p}_i$ and $\mathbf{p}_j$, are closer than a specified distance threshold T_c

$$\|\mathbf{p}_i - \mathbf{p}_j\| \leq T_c \quad \forall \quad (i \in S_i, j \in S_j) \tag{29}$$

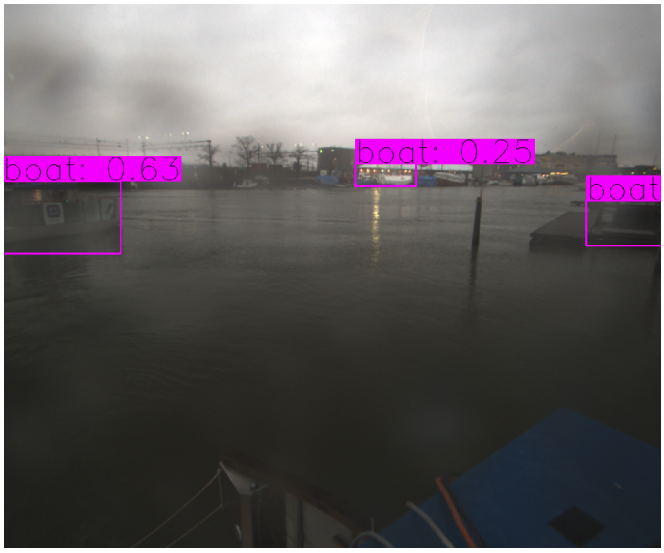
where S_i is the set of indexes for points in the cluster i. One might assume that a similar threshold as the maximum distance between generated points along the bottom bounding box would suffice, however, this does not take into account detection and navigation noise. At longer ranges, a few pixels of detection noise might induce a position difference of multiple meters. Due to this, the distance threshold has been set to $T_c = 3m$ in this work. For other platforms utilizing different detectors and navigation systems, this parameter might require further tuning. The output of the camera fusion process is visualized in Fig. 7.

1) MULTI-SPECTRAL FUSION

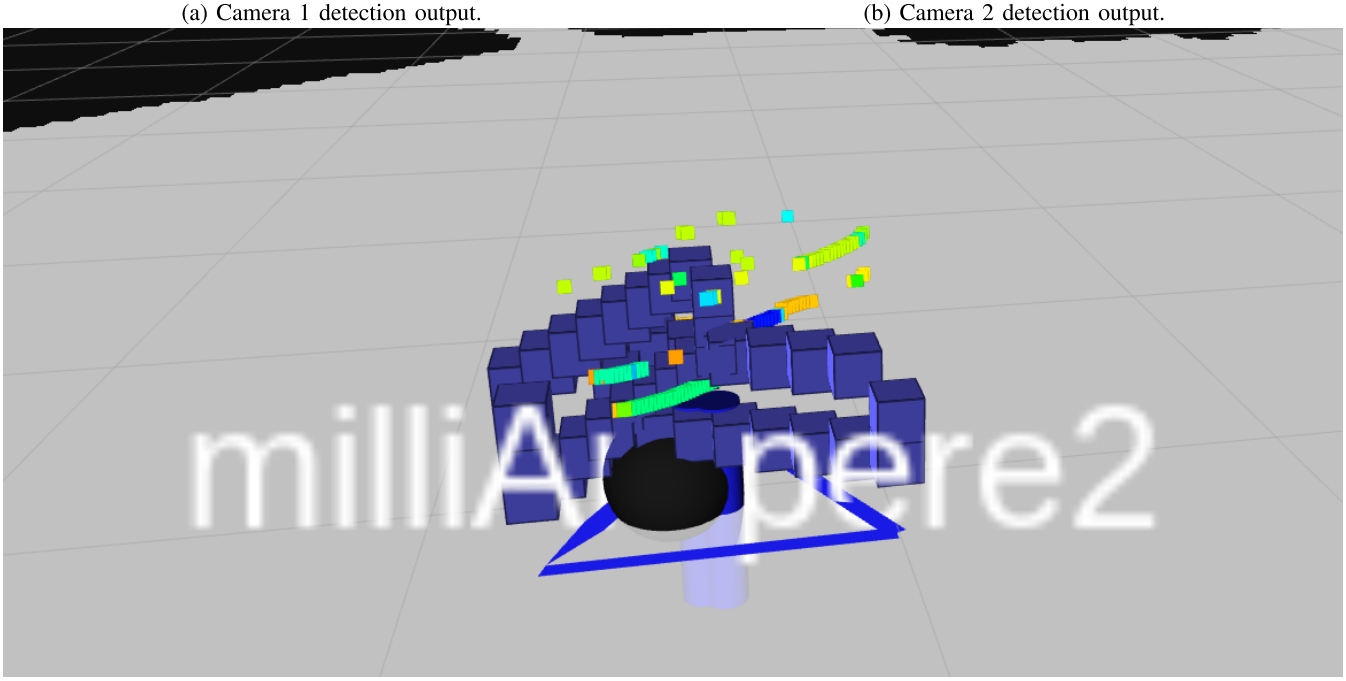
Another benefit of this approach is that measurements from imaging sensors sensitive to other spectral bands, such as infrared, can be fused into a single, more robust, detection. EO cameras enjoy a significant price and resolution advantage over most IR cameras. They are however more sensitive to conditions such as fog and scene illumination. Infrared



(a) Camera 1 detection output.



(b) Camera 2 detection output.



(c) Position estimates of the two bounding boxes, blue squares, and the resulting cluster outline, blue line with cylindrical center marking, overlayed on lidar data (colored dots). Ground truth given by black dot with text.

FIGURE 7. Fusion of detections from a target partially visible in two camera frames.

cameras, being sensitive to thermal radiation, will yield more consistent performance across different conditions. Introducing infrared detections in the camera fusion process could therefore yield a more robust sensing system and would in practice result in a pseudo-multi-spectral camera system realized through sensor fusion.

F. LIDAR BENCHMARK PIPELINE

The lidar benchmark pipeline utilizes the same land filtering and clustering algorithm as described above. Lidar point clouds, Fig. 9, are first converted from the lidar frame l to a local NED frame according to

$$\mathbf{p}_i^w = \mathbf{R}_l^w \mathbf{p}_i^l + \mathbf{t}_l^w. \tag{30}$$

Occupancy grid filtering is then applied to remove land returns and the resulting point cloud is then clustered to generate a single measurement per target. This pipeline is described in-depth in [1].

V. TRACKING SYSTEM

The tracking system used in this work is a single-sensor, single-model version of the multi-sensor VIMMJIPDA.

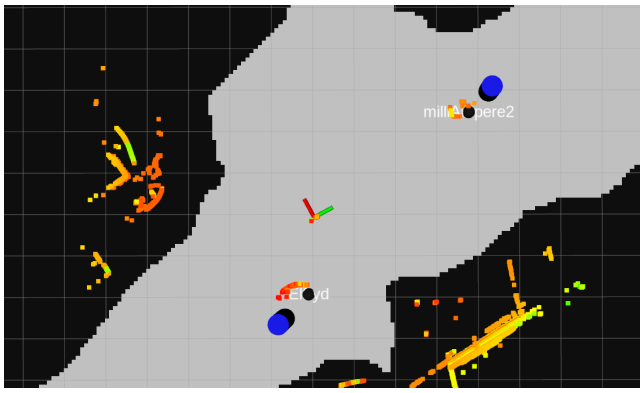


FIGURE 8. Multi-camera fusion output (blue cylinders) with GPS ground truth (black spheres and text) and lidar point cloud (colored squares) overlayed on occupancy grid.

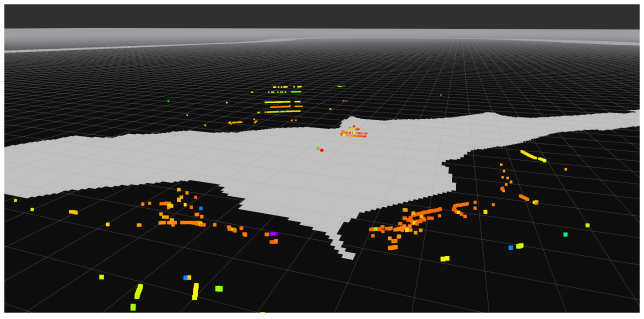


FIGURE 9. Lidar point cloud, shown as colored squares, and land occupancy grid. Square color is given by return intensity.

Algorithm 1 Detection Model

```
point_clouds ← empty list
cameras ← list of active cameras
for all camera in cameras do
    bboxes ← detect(image)
    point_cloud ← range_estimation(bboxes)
    point_clouds[camera] ← point_cloud
end for
fused_cloud ← aggregate_pcl(point_clouds)
filtered_cloud ← occ_grid_filter(fused_cloud)
detections ← cluster_pcl(filtered_cloud)
```

The VIMMJIPDA [29] multi-target tracker is a modern formulation of the Markov-chain two JIPDA with Interactive Multiple Models (IMM). The multi-sensor VIMMJIPDA, described in [1], is a multi-sensor extension of the VIMM-JIPDA with range-dependent sensor properties to better support heterogeneous sensor fusion. In this section, we describe the motion and sensor models used in this work leaving the complete derivations to [1] and [29].

A. MOTION MODEL

A common motion model in the field of target tracking is the constant velocity (CV) model. This assumes target velocities are constant, modeling acceleration as a Gaussian white noise process with zero mean. Target states are given by $\mathbf{x} = [p_x, p_y, v_x, v_y]$ where p and v are positions and velocities in a NED reference frame. For continuous time applications, the model is given by the equation

$$\dot{\mathbf{x}} = \mathbf{A}\mathbf{x} + \mathbf{G}\mathbf{n}. \tag{31}$$

Target process noise, modeling acceleration, is given by $\mathbf{n}$. This noise is assumed to be white with diagonal covariance, described by

$$\mathbf{n} \sim \mathcal{N}(\mathbf{0}, \mathbf{D}\delta(t-\tau)) \qquad \mathbf{D} = \begin{bmatrix} \sigma_a^2 & 0 \\ 0 & \sigma_a^2 \end{bmatrix} \tag{32}$$

where σ_a describes the typical acceleration of the target. The matrices $\mathbf{A}$ and $\mathbf{G}$ are given by

$$\mathbf{A} = \begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \qquad \mathbf{G} = \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 1 & 0 \\ 0 & 1 \end{bmatrix}. \tag{33}$$

For discrete-time applications, this model is discretized as

$$\mathbf{x}_k = \mathbf{F}\mathbf{x}_{k-1} + \mathbf{v}_k \qquad \mathbf{v}_k \sim \mathcal{N}(0, \mathbf{Q}) \tag{34}$$

where $\mathbf{F}$ is the state transition matrix, $\mathbf{x}_k$ the state at time-step k and $\mathbf{v}_k$ the discretized process noise with covariance $\mathbf{Q}$.

B. LIDAR SENSOR MODEL

The lidar does provide 3-dimensional sensor data in the form of point clouds, however, both ownship and target motion is constrained to the ocean plane and elevation data can therefore safely be discarded. Points in the point cloud are given in Cartesian coordinates in a sensor-fixed frame based on the range of the return, measured by time of flight, and the angle of the return, measured by the receiver rotation. This gives rise to a polar measurement model described by

$$f_z^l(\mathbf{x}_k) = \begin{bmatrix} \sqrt{p_x^2 + p_y^2} \\ \arctan(p_y/p_x) \end{bmatrix} + \mathbf{w}_k^l \qquad \mathbf{w}_k^l \sim \mathcal{N}(0, \mathbf{R}^l) \tag{35}$$

where f_z is the measurement function and w_k the sensor noise for the lidar l with covariance matrix $\mathbf{R}^l$. Due to the internal conversion from polar/spherical coordinates to Cartesian coordinates, this would require the clustered detections from the lidar pipeline to be converted back to polar coordinates. Instead, the following sensor model is used

$$f_z^l(\mathbf{x}_k) = \begin{bmatrix} p_x \\ p_y \end{bmatrix} + \mathbf{w}_k^l \qquad \mathbf{w}_k^l \sim \mathcal{N}(0, \mathbf{J}\mathbf{R}^l\mathbf{J}^{\mathrm{T}}) \tag{36}$$

where $\mathbf{J}$ is the Jacobian of the polar to Cartesian conversion of the measurement and $\mathbf{R}$ is the measurement noise in polar coordinates.

C. CAMERA SENSOR MODEL

After the clustering-based camera fusion is applied the camera detection pipeline outputs Cartesian detections. This yields the same measurement function as the lidar:

$$f_z^c(\mathbf{x}_k) = \begin{bmatrix} p_x \\ p_y \end{bmatrix} + \mathbf{w}_k^c. \tag{37}$$

D. MEASUREMENT NOISE

Ideally one would define the measurement noise in pixel coordinates as this is where the actual detection takes place. In the image plane, this measurement model takes the form of

$$f_z^p(\mathbf{x}_k^p) = \begin{bmatrix} x_k^p \\ y_k^p \end{bmatrix} + \mathbf{w}_k^p \qquad \mathbf{w}_k^p \sim \mathcal{N}(0, \mathbf{R}_p)). \tag{38}$$

where x^p and y^p are the pixel coordinates, $\mathbf{w}_k^p$ the measurement noise with pixel covariance $\mathbf{R}_p$ and $\mathbf{x}_k^p$ the target state in pixel coordinates obtained through the pinhole camera model. One issue with this approach is that when random variables, such as the pixel measurements, are transformed using a non-linear function such as the reverse pinhole projection of (14) we might introduce a bias in the transformed variable. In the case of camera-based georeferencing, the image plane will be roughly orthogonal to the ocean plane we project to and from. The area in the ocean plane that a single pixel covers will therefore depend on the distance from the camera to the point in question.

One approach to verify the presence, or absence, of a bias, is to approximate the transformed distribution using a Taylor series expansion of the transformed distribution. The distribution p of a pixel position $\mathbf{x}^p$ is given by

$$p \sim \mathcal{N}(\bar{\mathbf{x}}^p, \mathbf{R}_p) \tag{39}$$

Denoting the reverse pinhole transform as f_p^{-1}, the transformed distribution is given by

$$f_p^{-1}(\mathcal{N}(\bar{\mathbf{x}}^p, \mathbf{R}_p)) = f_p^{-1}(\mathbf{x}^p + \mathbf{w}_k^p). \tag{40}$$

Limiting ourselves to the first two degrees for simplicity, the Taylor series expansion of this distribution around the point $\mathbf{x}^p$ is

$$\begin{aligned} &f_p^{-1}(\mathbf{x}^p) + \nabla f_p^{-1}(\mathbf{x}^p)(\mathbf{x}^p - \bar{\mathbf{x}}^p) \\ &\quad + \frac{1}{2}(\mathbf{x}^p - \bar{\mathbf{x}}^p)^\intercal \nabla^2 f_p^{-1}(\mathbf{x}^p)(\mathbf{x}^p - \bar{\mathbf{x}}^p). \end{aligned} \tag{41}$$

This approximation has the mean value μ^T of

$$\begin{aligned} \mu^T &= \mathrm{E}[f_p^{-1}(\mathbf{x}^p) + \nabla f_p^{-1}(\mathbf{x}^p)(\mathbf{x}^p - \bar{\mathbf{x}}^p) \\ &\quad + \frac{1}{2}(\mathbf{x}^p - \bar{\mathbf{x}}^p)^\intercal \nabla^2 f_p^{-1}(\mathbf{x}^p)(\mathbf{x}^p - \bar{\mathbf{x}}^p)] \\ &= f_p^{-1}(\mathbf{x}^p) + \mathrm{E}[\nabla f_p^{-1}(\mathbf{x}^p)(\mathbf{x}^p - \bar{\mathbf{x}}^p)] \\ &\quad + \frac{1}{2}\mathrm{E}[(\mathbf{x}^p - \bar{\mathbf{x}}^p)^\intercal \nabla^2 f_p^{-1}(\mathbf{x}^p)(\mathbf{x}^p - \bar{\mathbf{x}}^p)] \\ &= f_p^{-1}(\mathbf{x}^p) + \begin{bmatrix} \mathrm{tr}(\mathbf{H}1(f_p^{-1}(\mathbf{x}^p))\mathbf{R}_p) \\ \mathrm{tr}(\mathbf{H}2(f_p^{-1}(\mathbf{x}^p))\mathbf{R}_p) \end{bmatrix} \end{aligned} \tag{42}$$

where $\mathbf{H}i$ is the Hessian of the i-th output of a function. Using just a second-order approximation it is already clear that the transformed distribution has a bias in its expected value. To illustrate this we will consider a detection of a target at a range of 100m directly in front of the camera, $\mathbf{x}^w = [100, 0]^\intercal$. Applying the pinhole model (10) to this target t results in the pixel coordinates

$$\mathbf{x}^p = \begin{bmatrix} 610 \\ 709 \end{bmatrix}. \tag{43}$$

In the pixel plane, the probability of a noise-induced vertical offset is equal in both directions assuming Gaussian noise. For an offset of ± 5 pixels vertically the resulting position estimates obtained through the reverse pinhole model, f_p^{-1}, are roughly

$$\mathbf{x}_1^w = f_p^{-1}(\begin{bmatrix} 610 \\ 714 \end{bmatrix}) = \begin{bmatrix} 89 \\ 0 \end{bmatrix} \tag{44}$$

$$\mathbf{x}_2^w = f_p^{-1}(\begin{bmatrix} 610 \\ 704 \end{bmatrix}) = \begin{bmatrix} 116.5 \\ 0 \end{bmatrix} \tag{45}$$

The first pixel position has an offset of 11m while the second has an offset of 16.5m, an increase of 5.5m. This effect also induces the distribution bias observed in (42) when the Gaussian pixel noise distribution is projected onto the ocean surface. Fig. 10 shows the contour lines of this distribution generated using a diagonal pixel covariance of

$$\mathbf{R}_p = \begin{bmatrix} 207.446 & 0 \\ 0 & 66.3718 \end{bmatrix} \tag{46}$$

which was found by sampling 60 image detections from the dataset. This covariance is also used to generate any additional distribution figures in this section.

The resulting distribution has a bias in its mean of nearly 6m along the range axis, estimated by random sampling, compared to the true target state. Comparing the expected (mean) values $\bar{\mathbf{x}}^w$ of the actual (sampled, 10^6 samples) and Taylor approximated distributions with the true position (47)

$$\begin{aligned} \bar{\mathbf{x}}_{sampled}^w &= \begin{bmatrix} 105.55 \\ 0.00 \end{bmatrix} \\ \bar{\mathbf{x}}_{Taylor}^w &= \begin{bmatrix} 104.66 \\ 0.00 \end{bmatrix} \\ \mathbf{x}^w &= \begin{bmatrix} 100.00 \\ 0.00 \end{bmatrix} \end{aligned} \tag{47}$$

reveals that while close to the sample mean, a second-order Taylor approximation is still not able fully capture the true moments of the transformed distribution due to the high non-linearity of the transform.

Examining the cross-section of the sampled distribution along the x-axis (range), Fig. 11, also reveals that the distribution is no longer symmetric but has a significant tail which results in the observed mean bias. This non-symmetricity also implies the distribution is no longer Gaussian, however, the distribution is still unimodal and a Gaussian approximation might therefore still yield sufficient performance. Along the y-axis, Fig. 12, the true distribution has a form that is much closer to the Gaussian measurement assumption of the JIPDA tracker with no observed bias in the mean.

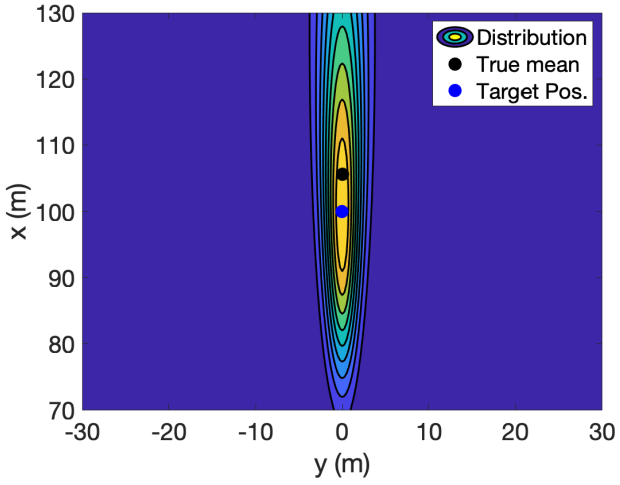


FIGURE 10. Contour lines of true measurement distribution.

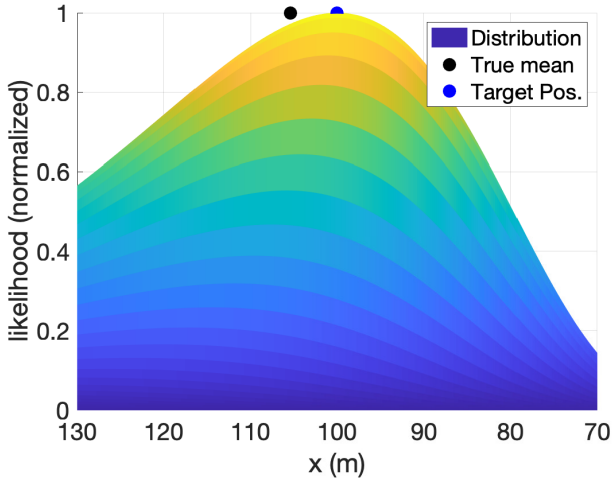


FIGURE 11. Cross section of true measurement distribution, x-axis.

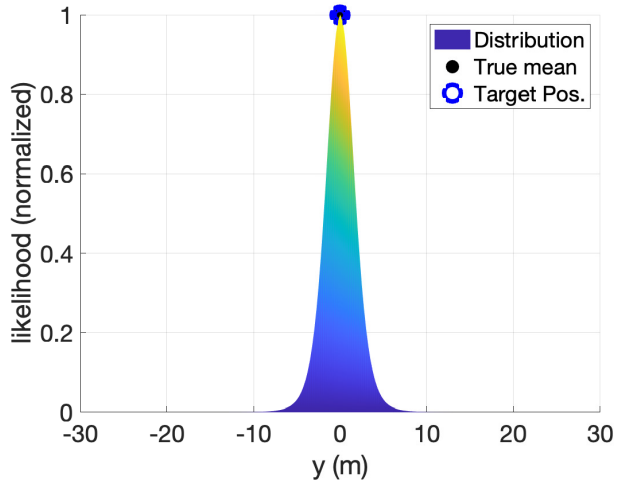


FIGURE 12. Cross section of true measurement distribution, y-axis.

1) DISTRIBUTION LINEARIZATION

Assuming $\mathbf{w}_k^c$ is Gaussian white, the simplest solution that retains the pixel parametrization of noise is to linearize around the expected pixel measurement, given by the target

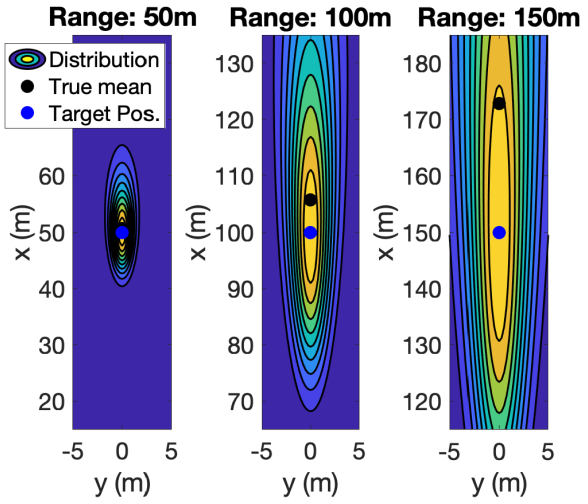


FIGURE 13. Contour lines of true distribution at ranges 50m, 100m and 150m.

state, and then convert the pixel noise to Cartesian coordinates using a Jacobian transform given by

$$\mathbf{w}_k^c \sim \mathcal{N}(0, \mathbf{J}\mathbf{R}_p\mathbf{J}^\intercal) \tag{48}$$

where $\mathbf{R}^c$ is the measurement noise in pixel coordinates and $\mathbf{J}$ is the Jacobian of the pixel-to-Cartesian conversion for the measurement as done in [20]. Deriving an analytic expression of the partial derivatives of the reverse pinhole equations (14) is possible, however, the resulting expression is long and numeric calculation might therefore be more attractive.

This approach increases the complexity of the measurement model compared to a range/bearing or Cartesian parametrization of noise and requires the tracker to have knowledge of the camera matrix $\mathbf{C}$. It does however allow for a more accurate description of sensor uncertainty. Close to the camera, a single pixel will only cover a small area, further away this area increases. Describing uncertainty in pixel coordinates and then converting to Cartesian will compensate for this, increasing Cartesian uncertainty with the target range. This approach does have some drawbacks requiring the tracker to both be aware of the camera's extrinsic as well as intrinsic parameters to calculate the required Jacobian matrices.

Comparing the resulting distribution and cross sections, Figs. 14, 15 and 16 also reveal that this does not account for the observed mean bias in the true distribution (47). The distribution comparison at various ranges shown in Fig. 13 reveals that at shorter ranges this bias can be negligible, however, at a range of 150m this bias grows significantly where linearization induces an error of roughly 25m which will significantly affect the accuracy of the tracking estimates. One could argue that in this particular case with a small, maneuverable ownship the actual impact on autonomous operations would be negligible. However, for larger, less maneuverable vessels or maritime surveillance purposes where even longer-range detections are typically required, the effects

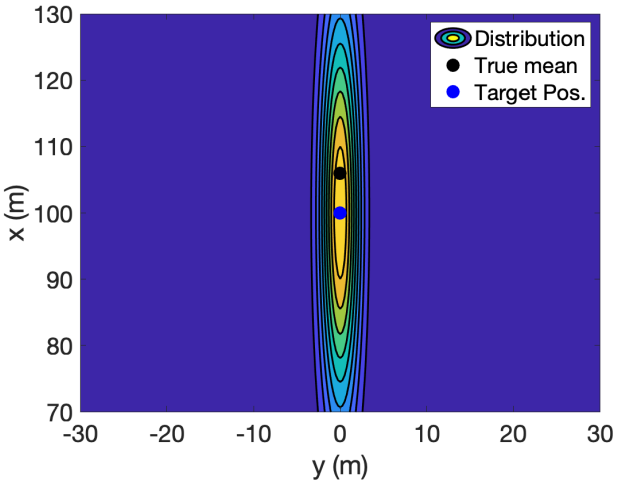


FIGURE 14. Contour lines of Jacobian transformed measurement distribution.

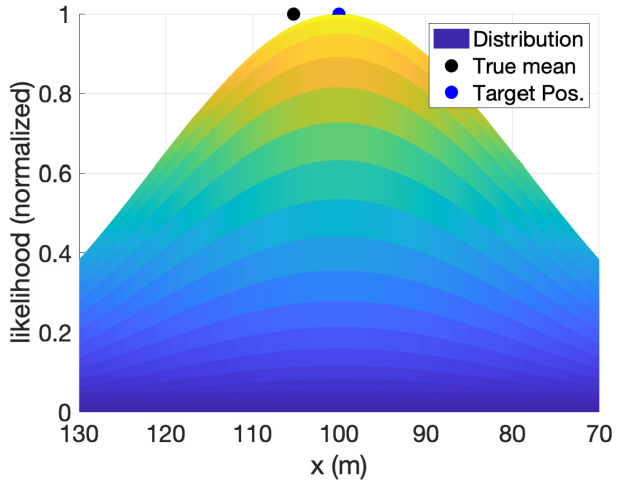


FIGURE 15. Cross section of Jacobian measurement distribution, x-axis.

would be much larger. Using the Taylor approximation mean value of (42) would remove much of this bias, however, the required Hessian matrices do introduce additional complexities.

Another potential issue with linearization is the consistency of the transformed covariance estimates. An estimate of covariance is consistent if the inequality

$$\mathbf{R}_e - \mathbf{R}_a \geq 0 \tag{49}$$

holds where e and a signify the estimated and actual covariance of the transformed random variable. Using the same target position and pixel covariance as previously, we find that the actual (sampled) and linearized covariances are

$$\mathbf{R}_l = \mathbf{J}\mathbf{R}_p\mathbf{J}^\mathsf{T} = \begin{bmatrix} 469.5 & 0.0 \\ 0.0 & 2.4 \end{bmatrix}$$
$$\mathbf{R}_a = \begin{bmatrix} 808.3 & 0.1 \\ 0.1 & 2.9 \end{bmatrix}. \tag{50}$$

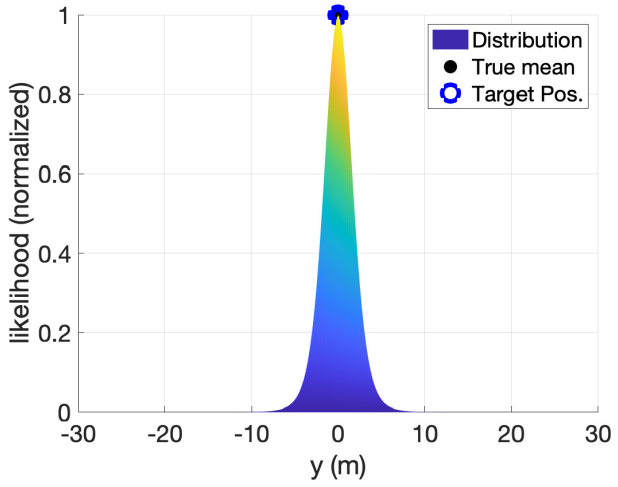


FIGURE 16. Cross section of Jacobian measurement distribution, y-axis.

The linearized covariance estimate is inconsistent along both axes which will cause the tracker to place greater weight on the already biased sensor measurements causing further divergences. Linearizing around (42) reduces this error slightly, yielding the still inconsistent estimate of

$$\mathbf{R}_e = \mathbf{J}\mathbf{R}_p\mathbf{J}^\mathsf{T} = \begin{bmatrix} 582.2 & 0.0 \\ 0.0 & 2.7 \end{bmatrix}. \tag{51}$$

Consistent covariance estimates will therefore require the addition of stabilizing noise. In this case increasing the base pixel covariance by a factor of 1.7 yields consistent estimates when linearizing around the target position. Linearization around (42) requires a slightly lower scaling of 1.4.

2) SCALED RANGE/BEARING NOISE

In contrast, a simple range/bearing parametrization of sensor noise avoids this complication. An examination of the true measurement distribution at various ranges, Fig. 13, reveals that uncertainty grows along both axes with increased range. A range/bearing parametrization, once converted to Cartesian coordinates, will experience similar growth in uncertainty along its bearing axis, however, the range uncertainty is constant which is not ideal. An alternate approach is to model measurement uncertainty as a range-scaled range-bearing covariance where an increased target range yields a corresponding increased range-bearing covariance. Target ranges are easily calculable based on tracking output requiring only the addition of a pre-specified covariance scaling function, f_R, resulting in

$$\mathbf{w}_k^c(\mathbf{x}_k^t) \sim \mathcal{N}(0, f_R(\mathbf{x}_k^t)). \tag{52}$$

for the target t at time k. This is however not investigated any further in this work.

3) UNSCENTED TRANSFORM

The Unscented transform [30], perhaps most known for its usage in the Unscented Kalman filter [31], was developed

as a way of estimating the outcome of a non-linear transformation of a distribution. Suppose we have a random variable, $\mathbf{x}$, with mean μ_x and covariance $\mathbf{P}_x$ which undergoes a non-linear transform resulting in $\mathbf{y} = f(\mathbf{x})$ which has the mean μ_y and covariance $\mathbf{P}_y$. When such a transform is part of the system models in an Extended Kalman filter-based system, the resulting distribution must be linearized. However, as observed previously, linearizing around the expected value can introduce both inconsistent covariance estimates and biases in the distribution mean.

The Unscented transform is an attempt to rectify these issues, selecting a more representative linearization point while reducing implementation complexity compared to the EKF. Instead of computing or approximating the Jacobian matrix of the transformation, the Unscented transform is based around sampling a set of sigma points with identical sample mean and covariance as the original distribution, μ_x and $\mathbf{P}_x$. The non-linear transform $f(\mathbf{x})$ is then applied to each point and the results are then weighted and combined to yield an estimate of the transformed mean and covariance, μ_y and $\mathbf{P}_y$.

While several methods have been developed to sample these sigma points [32], [33], [34], the original method described in [30] is based around sampling $2n+1$ points for an n-dimensional variable. Denoting $\mathbf{x}_i$ as the i-th sigma point, these points are given by

$$\begin{aligned} \mathbf{x}_0 &= \mu_x \\ \mathbf{x}_i &= \mu_x + \left(\sqrt{(n+\kappa)\mathbf{P}_x}\right)_i \qquad i \in \mathbb{Z},\ 1 < i \leq n \\ \mathbf{x}_{i+n} &= \mu_x - \left(\sqrt{(n+\kappa)\mathbf{P}_x}\right)_i \qquad i \in \mathbb{Z},\ 1 < i \leq n \end{aligned} \tag{53}$$

where $\left(\sqrt{(n+\kappa)\mathbf{P}_x}\right)_i$ is the i-th column of the matrix square root $\left(\sqrt{(n+\kappa)\mathbf{P}_x}\right)$ and κ a free parameter, typically $\kappa + n = 3$ for Gaussian distributions [31]. The associated weights, W_i, are given by

$$\begin{aligned} W_0 &= \frac{\kappa}{n+\kappa} \\ W_i &= \frac{1}{2(n+\kappa)} \\ W_{i+n} &= W_i \end{aligned} \tag{54}$$

The transformed sigma points, y_i, are then given by

$$\mathbf{y}_i = f(\mathbf{x}_i) \tag{55}$$

which are combined to yield the mean and covariance of the transformed distribution according to

$$\mu_y = \sum_{i=0}^{2n} W_i \mathbf{y}_i \tag{56}$$

and

$$\mathbf{P}_y = \sum_{i=0}^{2n} W_i (\mathbf{y}_i - \mu_y)(\mathbf{y}_i - \mu_y)^\mathsf{T}. \tag{57}$$

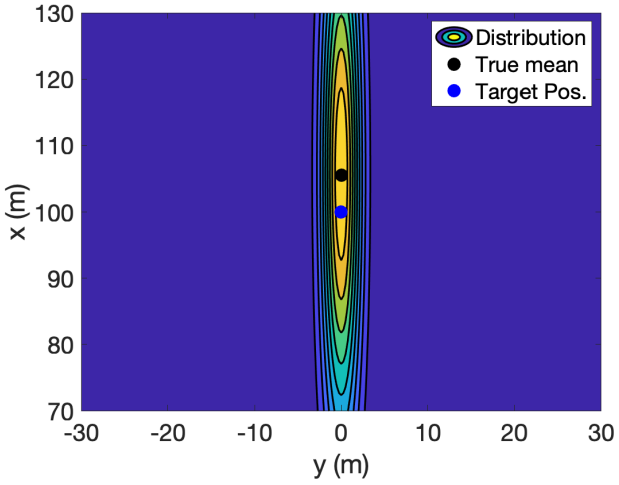


FIGURE 17. Contour lines of Unscented transformed measurement distribution.

Applying the Unscented transform to the reverse pinhole model (14) results in the distribution shown in Fig. 17. Compared to the linearized distribution, Fig. 14, the Unscented transform has virtually eliminated the mean bias,

$$\begin{aligned} \bar{\mathbf{x}}^w_{sampled} &= \begin{bmatrix} 105.55 \\ 0.00 \end{bmatrix} \\ \bar{\mathbf{x}}^w_{linearized} &= \begin{bmatrix} 100.00 \\ 0.00 \end{bmatrix} \\ \bar{\mathbf{x}}^w_{unscented} &= \begin{bmatrix} 105.70 \\ 0.00 \end{bmatrix}. \end{aligned} \tag{58}$$

This has also shifted the distribution along the x-axis, shown in Fig. 18, which better accounts for the large tail of the true distribution, Fig. 10. The y-axis cross-section, Fig. 19, has changed little.

Considering consistency, the Unscented transform delivers a marked improvement compared to linearization,

$$\begin{aligned} \mathbf{R}_U &= \begin{bmatrix} 802.6 & 0.0 \\ 0.0 & 2.5 \end{bmatrix} \\ \mathbf{R}_a &= \begin{bmatrix} 808.3 & 0.1 \\ 0.1 & 2.9 \end{bmatrix}. \end{aligned} \tag{59}$$

Along the x-axis, the standard deviation is underestimated by 0.1m resulting in a minor covariance difference. Along the y-axis the difference in standard deviation is almost identical, however, due to the low base value the difference in percentage is much greater.

Another benefit of the Unscented transform is that it can account for the effects of uncertainty in vessel pose if provided by the navigation system. For an active sensor such as the lidar, this uncertainty will have minor effects on the accuracy of the measurements as the sensor itself directly measures range and direction. In contrast, the reverse pinhole-based approach described in this work is much more dependent on accurate pose measurements. In its current implementation, *milliAmpere's* navigation system decouples position and pose estimation. Position estimates are provided

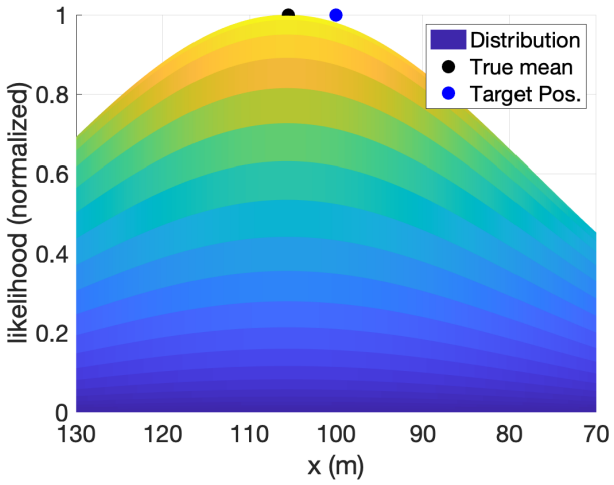


FIGURE 18. Cross section of Unscented measurement distribution, x-axis.

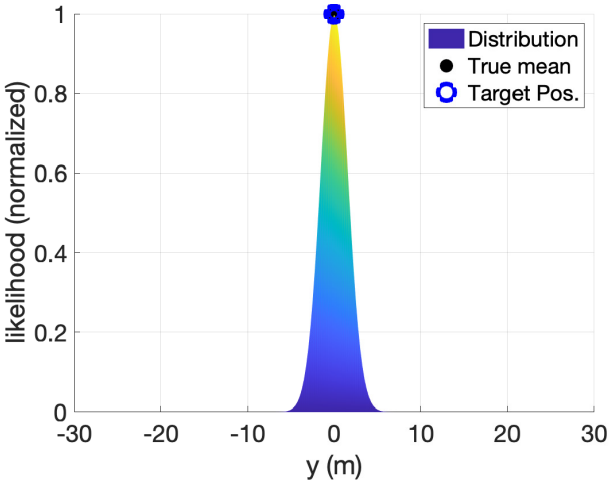


FIGURE 19. Cross section of Unscented measurement distribution, y-axis.

by the dual antenna GNSS compass along with vessel heading using RTK-corrected measurements which results in highly accurate positioning. Vessel pose however is provided by a simple $\alpha - \beta$ filter operating on IMU measurements which does not result in uncertainty measures. This filter also operates asynchronously with the cameras resulting in greater uncertainty in vessel pose.

VI. EXPERIMENTAL SETUP

This section describes the experimental setup used for the validation of the camera pipeline described in this work. Data collection took place in December in Trondheim, Norway. Due to the high latitude daylight intensity is quite weak in this period and is limited to roughly 5 hours per day. Combined with grey and overcast weather the lighting conditions were therefore quite challenging for the cameras.

A. REFERENCE TARGETS

Two reference targets were used in the data collection to enable multi-target scenarios with potential measurement ambiguity.

FIGURE 20. *Milliampere 2* during data collection.

1) TARGET 1

milliAmpere 2, Fig. 20, is a full-scale prototype of an autonomous urban passenger ferry designed from *milliAmpere 1*. *milliAmpere 2* is larger both in length and width to accommodate up to 12 passengers. Just like its smaller sibling, the vessel is highly maneuverable due to its four fixed-position thrusters, one in each corner. This also allows *milliAmpere 2* to reach a higher top speed compared to *milliAmpere 1*. *milliAmpere 2*'s sensor package is similar to *milliAmpere 1* with 8 electro-optical cameras, two in each corner, as well as a maritime radar and two lidars. *milliAmpere 2* operated under manual control during these experiments, however, an all-autonomous trial operation with passengers was conducted in September 2022 in the same area, Fig. 22.

2) TARGET 2

Elfryd, Fig. 21, is a small leisure craft with a low-profile wooden hull. This makes the vessel hard to detect both in radar and lidar data due to material properties and low cross-sectional area. *Elfryd* has been converted to electric power and is equipped with a single propeller powered by a battery pack located in the vessel hull. Speed-wise *Elfryd* is fairly slow and, due to its size and construction, is less maneuverable than most leisure crafts of similar size.

B. EXPERIMENTAL AREA

Shown in Fig. 22, the data collection took place in the canal between Brattøra, on the north side, and Ravnkloa, on the south side, in Trondheim, Norway. This is an urban environment with multiple jetties and mooring sites filled with vessels along both sides. Each scan will therefore contain multiple detections that should ideally be removed or labeled, making this an ideal stress test of the land filtering part of the pipeline.

C. SCENARIO 1

In this scenario, *milliAmpere 2* starts to the west in the canal traveling eastwards. *Elfryd* starts to the east traveling in the opposite direction, intersecting roughly in the middle. The

FIGURE 21. *Elfryd* as captured by *milliAmpere*'s sensor rig. The image has been post-processed to improve visual clarity.

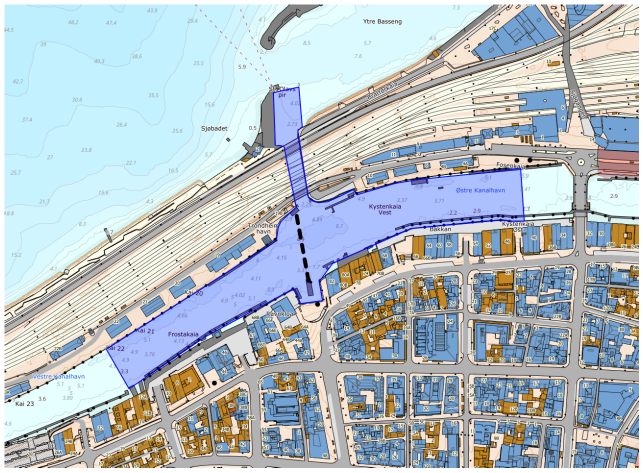


FIGURE 22. Experimental area, shown as a blue polygon, and the planned route of *milliAmpere 2*, shown as a black dotted line. Map data is provided by the Norwegian Mapping Authority (Kartverket).

ownship is stationary at Ravnkloa. A visualization is shown in Figure 23.

D. SCENARIO 2

This scenario is similar to scenario 1 with both targets traveling along the canal albeit from mirrored starting locations. The ownship starts at its docking location at the Brattøra side but travels out into the canal towards Ravnkloa when both targets have passed. A visualization is shown in Figure 24.

E. SCENARIO 3

This scenario is similar to scenario 1, however, instead of staying at the Ravnkloa dock, the ownship is stationary in the middle of the canal. Both targets maneuver around the ownship to their starboard, one on each side, intersecting with

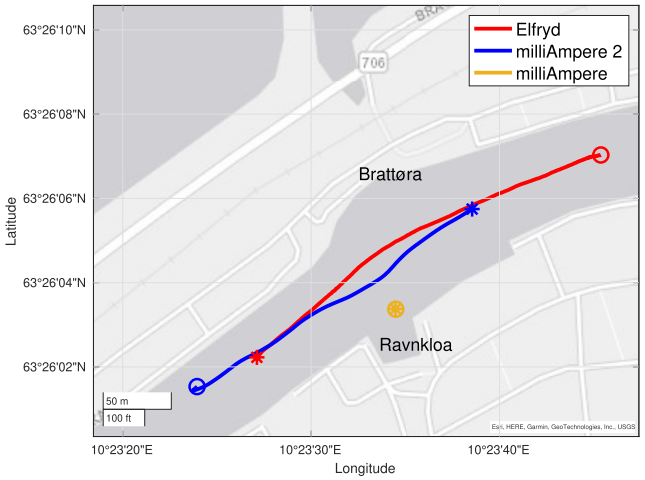


FIGURE 23. Scenario 1. Circles signify starting positions, stars end positions.

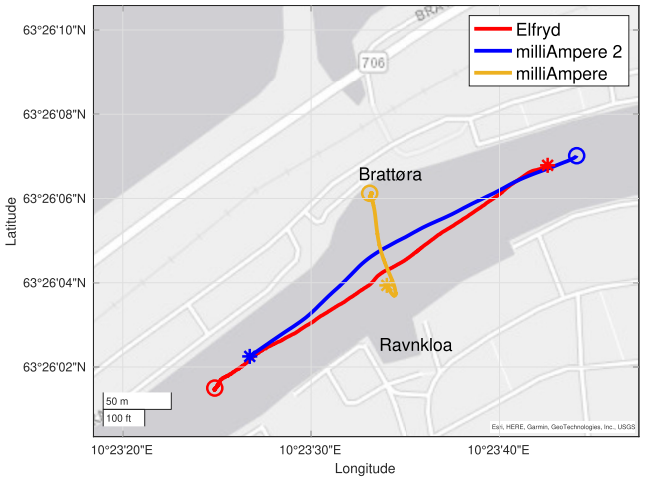


FIGURE 24. Scenario 2. Circles signify starting positions, stars end positions.

each other during this maneuver. A visualization is shown in Figure 25.

F. SCENARIO 4

This scenario is a repeat of scenario 3 but with ownship movement. Crossing from Ravnkloa to Brattøra, the ownship intersects both targets in the middle of the canal. Each target performs a maneuver to its starboard to avoid a collision. A visualization is shown in Figure 26.

G. SCENARIO 5

Once again each target starts on individual sides of the canal traveling straight towards the ownship which is stationary in the middle of the crossing. At a distance of roughly 50m, *milliAmpere 2* performs a stop to avoid a collision, letting *Elfryd* perform a starboard turn to avoid colliding with the ownship. Once passed, *milliAmpere 2* resumes its journey passing the ownship to the south as shown in Fig. 27.

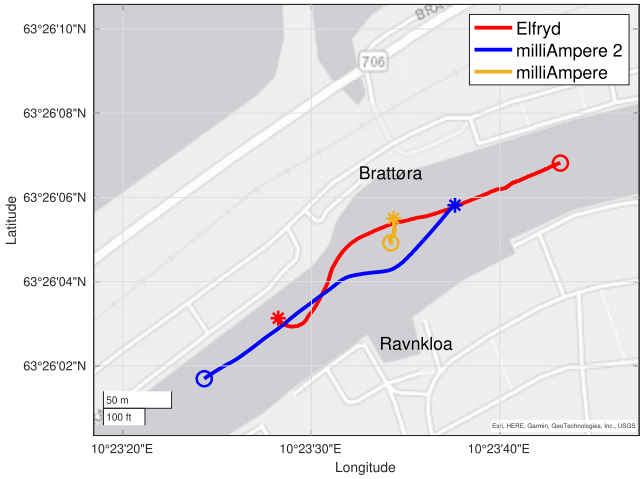


FIGURE 25. Scenario 3. Circles signify starting positions, stars end positions.

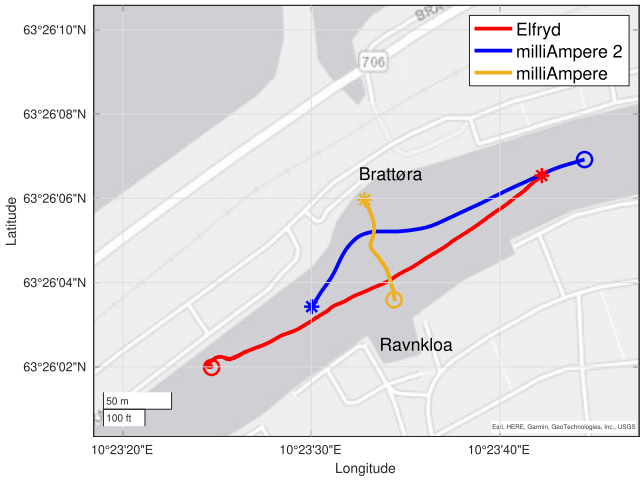


FIGURE 26. Scenario 4. Circles signify starting positions, stars end positions.

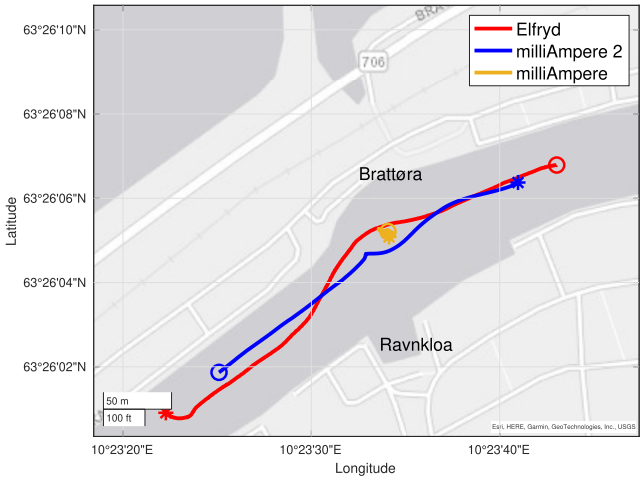


FIGURE 27. Scenario 5. Circles signify starting positions, stars end positions.

H. SCENARIO 6

This scenario, Fig. 28, repeats scenario 5 but with ownship movement. Starting from Brattøra, the ownship travels slowly

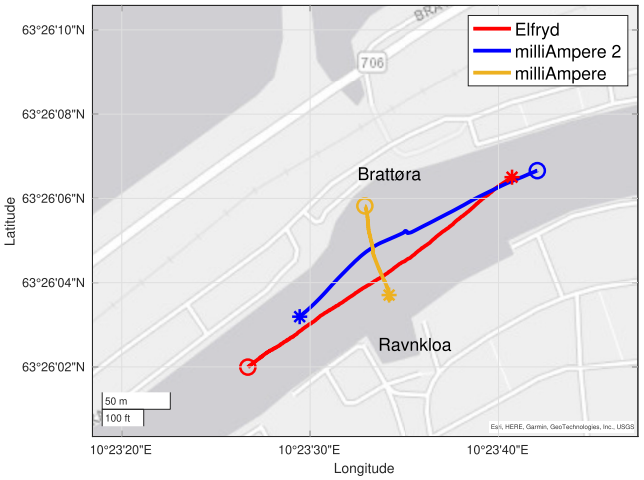


FIGURE 28. Scenario 6. Circles signify starting positions, stars end positions.

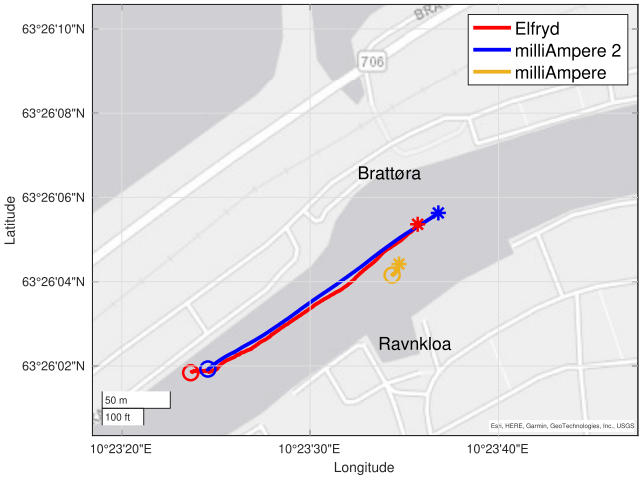


FIGURE 29. Scenario 7. Circles signify starting positions, stars end positions.

south towards Ravnkloa. Both targets intersect the ownship in the middle where they perform the same maneuvers as described in scenario 5.

I. SCENARIO 7

Starting to the west, both targets travel towards the ownship in a line with *Elfryd* obscured directly behind *milliAmpere 2*. A stationary ownship is then passed to the north by both targets as shown in Fig. 29.

VII. PERFORMANCE MEASURES

A series of performance measures covering multiple aspects of the system's performance is critical for an accurate evaluation of the performance of the detection system. In this work, we reuse the performance measures presented in [1] to simplify comparison with the complete sensor fusion system. Evaluations are performed automatically based on recorded GPS ground truth from both target vessels. Detections and

tracks are assigned to their closest target if the Cartesian distance is less than a distance threshold of 10m. This association is then used to compute the following metrics.

A. RMS ERROR

The root mean square error (RMSE) is a basic metric yielding information about the average error of the detection or tracking output. The square element of this metric punishes large outliers to a greater degree than a simple mean and is calculated according to

$$\text{RMSE} = \sqrt{\frac{1}{n}\sum_{i=1}^{n}|\mathbf{e}_i|^2}. \tag{60}$$

where $\mathbf{e}_i$ is a form of error, e.g. position or velocity. This metric is calculated for both detection errors and for tracking errors in terms of position

1) MEASUREMENT NOISE

Another important metric when it comes to sensor performance and system tuning is the measurement noise of a sensor. Higher measurement noise implies a greater degree of inaccuracy which has implications for how the tracking system weighs the prediction and measurement when updating states. Detection noise is reported as the measurement error covariance matrix, given by

$$\mathbf{R} = \frac{1}{n-1}\sum_{i=1}^{n}(\mathbf{z}_i - f_z(\mathbf{x}_j))(\mathbf{z}_i - f_z(\mathbf{x}_j))^T \tag{61}$$

where f_z is the measurement function of the sensor and n the set of indexes for measurements with an associated target. $\mathbf{z}_i$ and $\mathbf{x}_j$ are then individual measurements and their associated target states.

B. DETECTION PROBABILITY

Tuning the tracking system also requires the detection probability of a sensor which yields information about the likelihood of a target being detected in a single sensor scan. While useful for comparing sensor performance, this information is also used in the data association process of the tracker when computing association probabilities for the tracks and measurements.

In any received sensor scan a target is assumed to be detected if a measurement is assigned to it. We can then calculate the detection probability, P_D, as

$$P_D = \sum_{i=1}^{n}\frac{n_{det}^i}{n_{total}^i} \tag{62}$$

where n_{det} is the number of targets detected in scan i and n_{total} the number of targets present. We compute this information in range-specific bins to allow the tracking system to account for varying sensor performance across different ranges and to add additional information for evaluating the performance of the sensor system.

C. CLUTTER INTENSITY

The final sensor-specific tuning parameter is related to the number of false alarms, or clutter, that we can expect to be present in any single scan or region. In any received scan the unassociated measurements are assumed to be false alarms. We can then calculate the clutter intensity according to

$$\lambda = \frac{1}{\pi r_{max}^2}\frac{\sum_{k=1}^{n} m_k^{free}}{n_k} \tag{63}$$

where r_{max} is the maximum range of the sensor given by the evaluation of P_D or the spec sheet, n_k is the total number of time steps, and m_k^{free} is the number of unassociated measurements at time-step k. Similar to detection probability, false alarm intensities are also evaluated both as uniform across the entire sensor range and in range-specific bins for the active sensors.

D. ANEES

An important property of any estimator is the statistical consistency of the reported estimates. In this work, we use the average normalized estimation error squared (ANEES) which reports the relationship between the magnitude of the estimation errors and the covariance according to

$$\text{NEES}_k = \bar{\mathbf{x}}_i \mathbf{P}_i^{-1} \bar{\mathbf{x}}_i \tag{64}$$

where ANEES is given as the average of this across all time steps,

$$\text{ANEES} = \sum_{k=1}^{n}\text{NEES}_k. \tag{65}$$

E. ESTABLISHMENT LENGTH

Establishment length measures the amount of time from the start of a scenario to the establishment of a track on a target. We report this measure as the mean time across both targets and all datasets.

F. TRACK BREAKS

Once established, the tracking system should ideally keep valid tracks alive while within sensor range. This is not always the case, obscurement by other vessels or objects and a series of missed detections could reduce the existence probability of individual, valid tracks to a level that causes their termination. The track break performance measure reports this information in the form of both the number of track breaks and the total time of track breaks.

G. FALSE TRACKS

Another important aspect of track management is dealing with false tracks. False tracks are tracks that are not associated with a valid target, originating from clutter measurements. These tracks can interfere with the operation of an autonomous vessel in multiple ways. The motion planning part of the system could be induced to take unnecessary action due to the presence of these tracks and they also increase the

computational complexity of the tracking. Additionally, they could also prevent valid tracks from forming in their vicinity. False tracks are reported both as the total number of tracks and their total length in time.

H. GOSPA

A more modern performance measure for tracking systems is the general optimal subpattern assignment (GOSPA) [35]. This measure accounts for several aspects of the tracking process in a single measure, including position errors, false tracks, and missed targets. We define the set of tracks at time k as $\mathbf{X}_k = [\mathbf{x}_k^1, \dots \mathbf{x}_k^m]$ and the set of truths as $\mathbf{Y}_k = [\mathbf{y}_k^1, \dots \mathbf{y}_k^n]$. GOSPA is then given by

$$\text{GOSPA} = \left(\min_{\pi \in \prod_n} \sum_{i=1}^{m} d^{(c)}(\mathbf{x}_k^i, \mathbf{y}_k^{\pi(i)})^p + \frac{c^p}{\alpha}(n-m) \right)^{\frac{1}{p}} \tag{66}$$

where $\prod_n$ is the set of all permutations of $\{1, \dots, n\}$. $d(x, y)$ is a metric for track-truth distance and $d^{(c)}(x, y) = \min(d(x, y), c)$ the distance cut-off given by the parameter c. We use a Cartesian distance function with a cut-off of 10m. The rest of the parameters are set to $\alpha = p = 2$. The GOSPA metric exists both in a labeled and unlabeled form where the labeled form penalizes tracks switching from one target to another. In this work, we use the unlabeled GOSPA. This is motivated by the fact that basic collision avoidance does not care about track switching, only that a target is actually tracked. In more complex systems designed to track specific features such as target type or to be compliant with maritime maneuvering rules the labeled GOSPA is more appropriate. s

VIII. RESULTS

Based on the automatic evaluation system and performance measures presented in section VII and the datasets from section VI, the performance of the sensing and tracking system is examined in this section.

A. DETECTION PERFORMANCE

An evaluation of the detection performance of the various sensors has already been performed in [1]. This evaluation was, however, based on a different set of data with the cameras providing bearing measurements. This dataset uses a different set of targets where *milliAmpere 2* is quite far from a traditional boat shape while *Elfryd* has both a low cross-sectional area and a color that provides low contrast with the ocean surface. This might impart lower detection probability for both targets using the cameras and for *Elfryd* using the lidar. For this reason, we re-evaluate the detection performance of the sensors using this dataset.

From Fig. 30 we observe almost identical detection probability within the specified lidar max range of 100m with some differences starting to show close to this threshold. At further ranges, the lidar provides no detections while the cameras still provide sporadic measurements. In terms of clutter intensity, Fig. 31, the differences between the sensors are much greater.

TABLE 1. Detection performance. R is the diagonal elements of the sensor noise matrix (covariance), defined in range/bearing for the lidar and pixels for the cameras (range/bearing in parentheses). RMSE is in Cartesian coordinates. Detection probability is the average value within either 100m or the max detection range (in parentheses).

	Lidar	EO
R	$\begin{bmatrix} 12.46 \\ 0.0087 \end{bmatrix}$	$\begin{bmatrix} 207.45 \\ 66.37 \end{bmatrix}$ $\left(\begin{bmatrix} 54.17 \\ 0.0071 \end{bmatrix} \right)$
RMSE	6.40	9.39
λ	3.65e-5	3.11e-5
P_D	0.79 (0.66)	0.82 (0.49)

The lidar performs as one would expect from an active sensor. At closer ranges where the signal intensity is higher, we also receive a greater number of false alarms. At further ranges, this drops off. In contrast, the cameras have virtually no false detections at close ranges peaking instead in the mid ranges around 75m-100m.

This effect is due to instability in the range estimation, likely caused by poor navigation estimates of the vessel pose. In turn, this causes detections from moored and docked boats along the canal to oscillate in and out of the land map used to filter detections. One could argue that these are not strictly speaking false detections as they are potential valid targets. However, due to their large number, this would be very computationally expensive and could cause real-time performance degradation in the situational awareness system.

For pure detection accuracy, the lidar outperforms the cameras slightly. RMS detection error, Table 1, is 30% lower compared to the RGB cameras. Improving the navigation estimates should reduce this error somewhat, especially at mid to long-range. This difference is also reflected in the sensor noise covariance where the lidar has a much smaller uncertainty in range. Interestingly the RGB cameras seem to better be able to estimate target bearings. This might be due to the low spatial density of the lidar points at longer ranges. If only a target only generates a few returns the accuracy of the bearing measurement will be very dependent on the distribution of these points. Unless the target is perpendicular to the lidar it is likely that points will be unevenly distributed along the hull, potentially causing the observed effect. In contrast, the camera detector will for the most part generate a bounding box that covers the entire target regardless of orientation.

B. TRACKING PERFORMANCE

From a system-level perspective, the detection performance of the sensing system is less important than the resulting tracking performance it can deliver. Both path planning and collision avoidance systems operate on tracking estimates, both current and predicted, not pure detections. In this section we explore this tracking performance, comparing the lidar benchmark to the camera pipeline using the metrics presented in Section VII. The tracking system is based upon the sensor models presented in Section V with camera sensor noise described in pixel coordinates. Unscented transforms are then applied to yield the predicted Cartesian measurements and

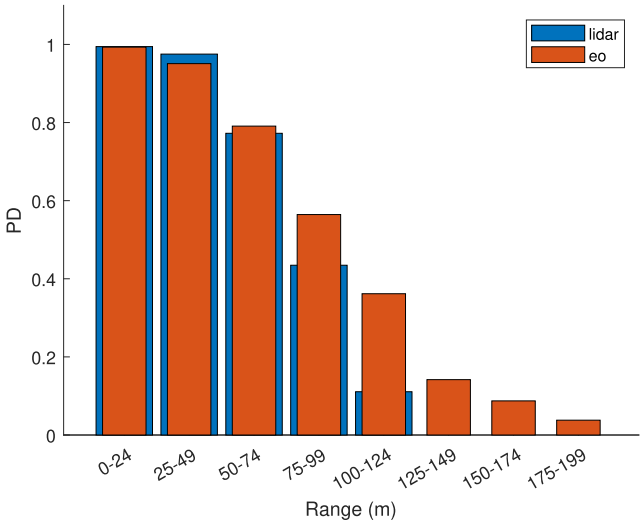


FIGURE 30. Detection probability.

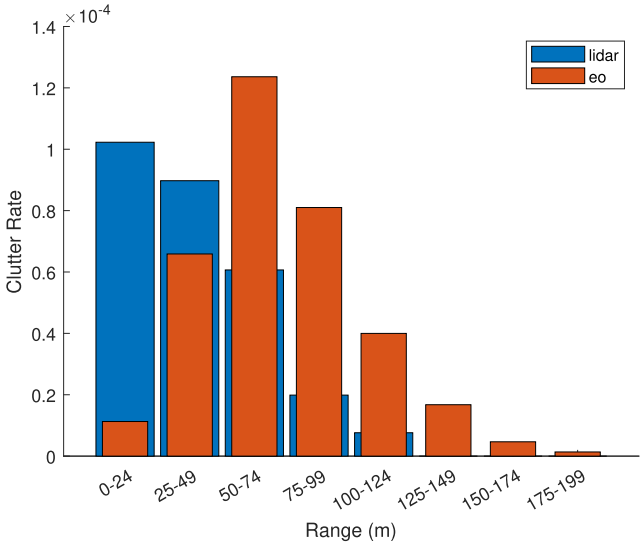


FIGURE 31. Clutter intensity.

covariances for the various tracks, also accounting for vessel pose uncertainty. Tuning parameters, shown in Appendix X, are for the most part based on [1] to avoid coloring the results with scenario-specific tuning. The only exception to this is the camera sensor noise and clutter intensities due to the change from bearing measurements to Cartesian measurements.

Starting with track management performance we find that both sensors are virtually identical when it comes to track establishment, Table 2. This result is not as expected based on the long-range detection performance, Fig. 30, which implies that track establishment lengths should be shorter for the cameras. A likely explanation is that the detections at these ranges are not stable enough to actually establish a valid track. Further tuning of the track initialization process might improve this performance somewhat, however, any track established at these ranges would have large uncertainties

TABLE 2. Tracking performance. Due to space constraints units have been excluded. Establishment lengths (Est.L) and break lengths (Break.L) are in seconds while position RMS error (posRMSE) is reported in meters per target. GOSPA is reported as RMS.

Sensors	Est.L	Break.L	posRMSE	GOSPA	ANEES
L	51.6	17.2	23.2;22.1	23.0	17.55
EO	53.8	30.1	11.5;12.9	26.9	95.46

Sensors	Num. False Tracks	False Track Length
L	65	1148
EO	268	2051

and therefore less useful for other parts of the autonomy system. Track break lengths are perhaps for the same reason longer with camera-based tracking compared to the lidar. This variation in detections at longer ranges can reduce track existence probability, potentially to a level where the track is terminated. False tracks are also an issue for the same reasons. Unsynchronized and noisy navigation estimates will cause larger variations in the detection ranges for the camera pipeline compared to the lidar. For docked boats this results in a position estimate that dips in and out of the land filtering area to a much greater extent, causing increased clutter intensity at medium ranges and a corresponding increase in false tracks.

On the other hand, the tracking performance of the cameras appears to be roughly equal to the lidar depending on the metric in question. For pure positioning accuracy, the cameras actually halve the RMS error compared to the lidar benchmark, reversing the result observed in the detection evaluation. This result is also not limited to a single target as one might expect considering the low cross-sectional area of *Elfryd*. Compared to previous evaluations of the lidar [1] we also observe a significant degradation in both consistency and accuracy on these datasets suggesting that further tuning work might be necessary. Statistical consistency, ANEES, is also poor. Both sensors underestimate the errors in the state estimates, the cameras to a much larger degree than the lidar. GOSPA, which accounts for both tracking accuracy as well as track management, is similar for both sensors with slightly better lidar performance.

IX. DISCUSSION AND FUTURE WORK

The camera detection pipeline has shown promising performance, offering performance that exceeds the lidar in several benchmarks. Accurate land filtering retains its status as an issue from [1], this time also applicable to camera detections. This issue is exacerbated by *milliAmpere's* pose estimates. Improved accuracy and camera synchronization can offer large performance benefits in this effect, reducing the number of clutter measurements and improving range estimates, especially at further ranges.

Statistical consistency also requires further work and is likely dependent on both improved tuning and improved detection performance. A Kalman filter-based navigation system will provide uncertainty estimates that can be used in the Unscented measurement transform. Combined with better

TABLE 3. Sensor tuning parameters. R is given as the diagonal elements of the covariance matrix, lidar in polar coordinates and EO in pixel.

	R
Lidar	$[33.7763\text{m}^2, 0.0054\text{rad}^2]$
EO	$[280, 60]\text{px}^2$

TABLE 4. Range dependent false alarm intensities (λ).

	Lidar	EO
0m-49m	4.39e-5	3.86e-6
50m-99m	1.06e-5	1.02e-4
100m-149m	8.16e-6	2.83e-5
150m-199m	0	3.01-6
200m-249m	0	0
250m-299m	0	0
300m-349m	0	0
349m-399m	0	0

pose estimates this should have a significant positive effect on both consistency and accuracy and reduce the number of false tracks. An alternative approach is to augment the noise model with fixed, stabilizing noise to improve consistency. This is more of an ad-hoc approach that increases the tuning complexity and might not generalize to other scenarios but it could reduce the observed inconsistencies. We briefly investigated this with additive range/bearing parametrized noise and while consistency improved significantly, both position RMSE and GOSPA suffered and it was therefore not included as part of this work.

Dynamic land filtering, as opposed to the current pre-determined static approach, is another aspect that could improve performance. The dynamic nature of the operating environment requires constant updates to the land map for optimal performance. Three medium-to-large vessels moored outside each other in a row, extending 10s of meters into the canal, might be gone the following day freeing up a large area for targets to maneuver in. Static land maps must thus weigh extending the map into valid areas to filter unwanted static detections from moored boats against the removal of potentially valid targets in these areas. A dynamic land map based on Simultaneous Localization and Mapping could be a solution to this.

A closed-loop collision avoidance experiment using *milliAmpere 2* will be reported in a forthcoming publication. *MilliAmpere 2* is equipped with a state-of-the-art navigation system synchronized with sensor readings, providing more accurate pose estimates for the cameras. On the other hand, the cameras are mounted lower which will increase measurement uncertainty compared to *milliAmpere 1*'s mounting location.

X. CONCLUSION

A novel detection pipeline for maritime target tracking with georeferencing-based range estimation and multi-camera fusion was described in this work. Real-world data evaluation showed comparable performance to a lidar benchmark across many performance metrics but with superior long-range detection performance. Several issues with the experimental platform were uncovered that could result in even better performance if fixed, however, even in its current state the pipeline offers value both as an additional sensor in a sensor fusion system and as a backup collision avoidance sensor in case of active sensor failures.

TABLE 5. Range dependent detection probabilities.

	Lidar	EO
0m-49m	0.99	0.77
50m-99m	0.96	0.77
100m-149m	0.79	0.86
150m-199m	0	0.66
200m-249m	0	0
250m-299m	0	0
300m-349m	0	0
349m-399m	0	0

TABLE 6. JIPDA tuning parameters.

σ_a	1.5	Process noise
g	3	Gate size
η_0	0.5	Init. exi. prob.
ϵ_0	1	Init. obs. prob.
T_{conf}	0.8	Track confirmation threshold
T_r	0.25	Track termination threshold
$\mathbf{F}_\eta$	$\begin{bmatrix} 0.99 & 0.01 \\ 0 & 1 \end{bmatrix}$	Exi. transition matrix
$\mathbf{F}_\epsilon$	$\begin{bmatrix} 0.9 & 0.1 \\ 0.48 & 0.52 \end{bmatrix}$	Obs. transition matrix
$\mathbf{v}_{max}$	10 m/s	Track init. max speed

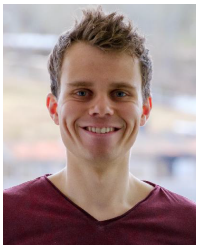
APPENDIX A TUNING PARAMETERS

The various tuning parameters used in this work are shown in Tables 3, 4, 5 and 6 and are based on the sensor evaluation from environment 1.

REFERENCES

[1] 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 Eng.*, vol. 252, May 2022, Art. no. 111168.
[2] D. C. Andrade, F. Bueno, F. R. Franco, R. A. Silva, J. H. Z. Neme, E. Margraf, W. T. Omoto, F. A. Farinelli, A. M. Tusset, S. Okida, M. M. D. Santos, A. Ventura, S. Carvalho, and R. D. S. Amaral, "A novel strategy for road lane detection and tracking based on a vehicle's forward monocular camera," *IEEE Trans. Intell. Transp. Syst.*, vol. 20, no. 4, pp. 1497–1507, Apr. 2019.
[3] X. Zhao, P. Sun, Z. Xu, H. Min, and H. Yu, "Fusion of 3D LiDAR and camera data for object detection in autonomous vehicle applications," *IEEE Sensors J.*, vol. 20, no. 9, pp. 4901–4913, May 2020.
[4] D. Hermann, R. Galeazzi, J. Andersen, and M. Blanke, "Smart sensor based obstacle detection for high-speed unmanned surface vehicle," in *Proc. 10th IFAC Conf. Manoeuvring Control Mar. Craft (MCMC)*, vol. 48, 2015, pp. 190–197.
[5] D. Cormack, I. Schlangen, J. R. Hopgood, and D. E. Clark, "Joint registration and fusion of an infrared camera and scanning radar in a maritime context," *IEEE Trans. Aerosp. Electron. Syst.*, vol. 56, no. 2, pp. 1357–1369, Apr. 2020.
[6] S. Giompapa, A. Farina, F. Gini, A. Graziano, and R. di Stefano, "Computer simulation of an integrated multi-sensor system for maritime border control," in *Proc. IEEE Radar Conf.*, Apr. 2007, pp. 308–313.
[7] 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," *J. Field Robot.*, vol. 27, no. 6, pp. 819–833, Nov. 2010.

[8] F. Schöller, M. Blanke, M. K. Plenge-Feidenhans, and L. Nalpantidis, "Vision-based object tracking in marine environments using features from neural network detections," in *Proc. 21st IFAC World Congr.*, 2020, vol. 53, no. 2, pp. 14517–14523.
[9] T. Brehard and J.-P. L. Cadre, "Closed-form posterior Cramer–Rao bounds for bearings-only tracking," *IEEE Trans. Aerosp. Electron. Syst.*, vol. 42, no. 4, pp. 1198–1223, Oct. 2006.
[10] J. M. C. Clark, R. B. Vinter, and M. M. Yaqoob, "Shifted Rayleigh filter: A new algorithm for bearings-only tracking," *IEEE Trans. Aerosp. Electron. Syst.*, vol. 43, no. 4, pp. 1373–1384, Oct. 2007.
[11] N. Peach, "Bearings-only tracking using a set of range-parameterised extended Kalman filters," *IEE Proc., Control Theory Appl.*, vol. 142, no. 1, pp. 73–80, 1995.
[12] Y. Kuznietsov, J. Stückler, and B. Leibe, "Semi-supervised deep learning for monocular depth map prediction," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 2215–2223.
[13] H. Fu, M. Gong, C. Wang, K. Batmanghelich, and D. Tao, "Deep ordinal regression network for monocular depth estimation," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 2002–2011.
[14] N. Lazaros, G. C. Sirakoulis, and A. Gasteratos, "Review of stereo vision algorithms: From software to hardware," *Int. J. Optomechatron.*, vol. 2, no. 4, pp. 435–462, 2008.
[15] A. Masoumian, H. A. Rashwan, J. Cristiano, M. S. Asif, and D. Puig, "Monocular depth estimation using deep learning: A review," *Sensors*, vol. 22, no. 14, p. 5353, Jul. 2022.
[16] J. L. Schönberger and J.-M. Frahm, "Structure-from-motion revisited," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 4104–4113.
[17] Y. Bar-Shalom and E. Tse, "Tracking in a cluttered environment with probabilistic data association," *Automatica*, vol. 11, no. 5, pp. 451–460, 1975.
[18] T. E. Fortmann, Y. Bar-Shalom, and M. Scheffe, "Multi-target tracking using joint probabilistic data association," in *Proc. 19th IEEE Conf. Decis. Control Including Symp. Adapt. Processes*, Dec. 1980, pp. 807–812.
[19] S. S. Blackman, "Multiple hypothesis tracking for multiple target tracking," *IEEE Aerosp. Electron. Syst. Mag.*, vol. 19, no. 1, pp. 5–18, Jan. 2004.
[20] O. K. Helgesen, E. F. Brekke, A. Stahl, and O. Engelhardtsen, "Low altitude georeferencing for imaging sensors in maritime tracking," in *Proc. 21st IFAC World Congr.*, 2020, vol. 53, no. 2, pp. 14476–14481.
[21] D. Musicki and R. Evans, "Joint integrated probabilistic data association: JIPDA," *IEEE Trans. Aerosp. Electron. Syst.*, vol. 40, no. 3, pp. 1093–1099, Jul. 2004.
[22] E. F. Brekke, E. Eide, B.-O. H. Eriksen, E. F. Wilthil, M. Breivik, E. Skjellaug, O. K. Helgesen, A. Lekkas, A. B. Martinsen, E. H. Thyri, T. Torben, E. Veitch, O. A. Alsos, and T. A. Johansen, "milliAmpere: An autonomous ferry prototype," in *Proc. 4th Int. Conf. Maritime Auto. Surf. Ships*, 2022, Art. no. 012029.
[23] B. E. Bayer, "Color imaging array," U.S. Patent 3 971 065, Jul. 20, 1976.
[24] K. Hirakawa and T. W. Parks, "Adaptive homogeneity-directed demosaicing algorithm," *IEEE Trans. Image Process.*, vol. 14, no. 3, pp. 360–369, Mar. 2005.
[25] Z. Zhang, "A flexible new technique for camera calibration," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 22, no. 11, pp. 1330–1334, Nov. 2000.
[26] A. Bochkovskiy, C. Wang, and H. M. Liao, "YOLOv4: Optimal speed and accuracy of object detection," *CoRR*, vol. abs/2004.10934, pp. 1–17, Apr. 2020.
[27] T. Lin, M. Maire, S. J. Belongie, L. D. Bourdev, R. B. Girshick, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick, "Microsoft COCO: Common objects in context," *CoRR*, vol. abs/1405.0312, pp. 1–15, Feb. 2014.
[28] E. F. Wilthil, A. L. Flåten, and E. F. Brekke, "A target tracking system for ASV collision avoidance based on the PDAF," in *Sensing and Control for Autonomous Vehicles*, vol. 474, P. Fossen and H. Nijmeijer, Eds. Ålesund, Norway: Springer, 2017, pp. 269–288.
[29] E. F. Brekke, A. G. Hem, and L.-C.-N. Tokle, "Multitarget tracking with multiple models and visibility: Derivation and verification on maritime radar data," *IEEE J. Ocean. Eng.*, vol. 46, no. 4, pp. 1272–1287, Oct. 2021.
[30] J. K. Uhlmann, "Dynamic map building and localization: New theoretical foundations," Ph.D. thesis, Dept. Eng. Sci., Univ. Oxford, Oxford, U.K., 1995.
[31] S. J. Julier and J. K. Uhlmann, "New extension of the Kalman filter to nonlinear systems," *Proc. SPIE*, vol. 3068, pp. 182–193, Apr. 1997.
[32] S. J. Julier and J. K. Uhlmann, "Reduced sigma point filters for the propagation of means and covariances through nonlinear transformations," in *Proc. Amer. Control Conf.*, vol. 2, 2002, pp. 887–892.
[33] Y. Cheng and Z. Liu, "Optimized selection of sigma points in the unscented Kalman filter," in *Proc. Int. Conf. Electr. Control Eng.*, Sep. 2011, pp. 3073–3075.
[34] D. Ebeigbe, T. Berry, M. M. Norton, A. J. Whalen, D. Simon, T. Sauer, and S. J. Schiff, "A generalized unscented transformation for probability distributions," 2021, *arXiv:2104.01958*.
[35] A. S. Rahmathullah, A. F. García-Fernández, and L. Svensson, "Generalized optimal sub-pattern assignment metric," in *Proc. 20th Int. Conf. Inf. Fusion (Fusion)*, Jul. 2017, pp. 1–8.



ØYSTEIN K. HELGESEN received the M.Sc. degree in cybernetics engineering from NTNU, Norway, in 2019, where he is currently pursuing the Ph.D. degree in cybernetics engineering. His research interests include sensor fusion and situational awareness for urban autonomous ferries as a part of the Autoferry project.

ANNETTE STAHL received the Ph.D. degree in applied mathematics (research field computer vision) from the Image and Pattern Analysis (IPA) Group, Department of Mathematics and Computer Science, University of Heidelberg, Germany, in 2009. She spent two years as a Postdoctoral Research Fellow with the School of Computing, Dublin City University (DCU), Ireland, and three years with the Department of Mathematical Sciences, NTNU, Norway. She was as a Researcher with the High-Performance Computing Group, NTNU, and SINTEF Ocean, Norway, where she was concerned with computer vision-based aquaculture applications. She is currently within the field of robotic vision targeting underwater, on sea surface, on land, in air, and space and indoor and industrial-related robotic applications. In 2016, she was awarded an Onsager Fellowship from NTNU's Research Excellence.

EDMUND F. BREKKE (Senior Member, IEEE) received the M.Sc. degree in industrial mathematics and the Ph.D. degree in cybernetics engineering from NTNU, Norway, in 2005 and 2010, respectively. His Ph.D. research covered methods for single-target tracking in clutter. After his Ph.D. studies, he was a Postdoctoral Research Fellow with the Acoustic Research Laboratory, NUS, Singapore. He was appointed as an Associate Professor in sensor fusion with NTNU, in 2014. His research interests include Bayesian estimation, with applications in target tracking and autonomous vehicles.