Internship report

Enhancing Situational Awareness for Maritime Autonomous Surface Ships (MASS) with RGB and LWIR camera fusion

Elodie Conan
ENSTA
elodie.conan@ensta.fr



Hosting institution
DTU Electro – Technical University of Denmark

Supervisor
Roberto Galeazzi

Mentors
Morgan Louédec, Jonathan Eichild Schmidt

Internship period
04/05/2026 – 04/09/2026

Abstract

Maritime autonomous surface ships (MASS) require reliable situational awareness, particularly in low-light and adverse weather conditions where visible-spectrum cameras may be limited. This internship, conducted within the CREA research group at DTU in Copenhagen, investigates the use of a long-wave infrared (LWIR) camera combined with a pair of RGB cameras for the detection, localisation, and tracking of boats from a fixed land-based platform.

Boats are detected and segmented in LWIR images using a YOLO11-seg model fine-tuned on the MassMIND dataset. The position of each boat is then estimated by intersecting the corresponding viewing ray associated with the detected waterline pixel with the sea plane. These estimates are fused with position estimates from an RGB stereo branch using a world-frame Kalman filter. The approach was evaluated on a sequence recorded in Copenhagen Harbour. The results demonstrate the feasibility of localising and tracking boats by combining thermal and RGB vision. The developed framework provides a foundation for future research at DTU towards further improving boat localisation through the integration of thermal cameras.

Résumé

Les navires de surface autonomes (MASS) nécessitent une perception fiable de leur environnement, notamment dans des conditions de faible luminosité et de météo défavorable, où les caméras opérant dans le spectre visible peuvent être limitées. Dans le cadre de ce stage, réalisé au sein du groupe de recherche CREA de la DTU à Copenhague, l'utilisation d'une caméra infrarouge thermique à ondes longues (LWIR), combinée à une paire de caméras RGB, est étudiée pour la détection, la localisation et le suivi de bateaux depuis une plateforme fixe située à terre.

Les bateaux sont détectés et segmentés dans les images LWIR à l'aide d'un modèle YOLO11-seg réentraîné sur le jeu de données MassMIND. La position de chaque bateau est ensuite estimée en calculant l'intersection entre le rayon de visée associé au pixel détecté sur la ligne de flottaison et le plan de la surface de la mer. Ces estimations sont fusionnées avec les estimations de position issues d'une branche de vision stéréo RGB au moyen d'un filtre de Kalman exprimé dans un repère monde. L'approche a été évaluée sur une séquence enregistrée dans le port de Copenhague. Les résultats démontrent la faisabilité de la localisation et du suivi de bateaux en combinant la vision thermique et la vision RGB. Le cadre développé constitue une base pour de futurs travaux de recherche à la DTU visant à améliorer la localisation des bateaux grâce à l'intégration de caméras thermiques.

Contents

Introduction **1**

1 Internship context **2**
1.1 Presentation of the research group . . . 2
1.2 Context of the project . . . 2
1.3 Presentation of the land platform . . . 2

2 Pipeline **3**
2.1 Detection . . . 4
2.2 Position estimation . . . 4
2.3 Fuse and track . . . 5

3 Detection **5**
3.1 Dataset preparation . . . 6
3.2 Training . . . 7
3.3 Evaluation . . . 8

4 Position estimation **9**
4.1 Calibration . . . 10
4.2 Sea-plane intersection . . . 11
4.3 Determining the waterline pixel coordinates . . . 13
4.4 Uncertainty . . . 14
4.5 Results . . . 15

5 Kalman filter **15**
5.1 Theory . . . 15
5.2 Results . . . 18

Conclusion **19**

Introduction

This report presents the four-month research internship I carried out at the Technical University of Denmark (DTU) during my second year of engineering studies at ENSTA, corresponding to the first year of a Master's degree.

The development of Maritime Autonomous Surface Ships (MASS) is opening new possibilities for maritime transportation, monitoring, and research. As ships become increasingly autonomous, their ability to perceive and understand their surrounding environment becomes essential for safe and reliable navigation. This situational awareness relies on sensor data to detect and localize surrounding vessels and obstacles.

In some applications, autonomous or remotely operated vessels can be monitored from a land-based station. In such a configuration, the operator needs reliable information about the maritime environment in order to supervise the vessel and react appropriately when necessary. Camera-based perception can provide valuable visual information, but its performance can be affected by environmental conditions such as poor visibility, low illumination, fog, rain, or sun glare. Using complementary sensing modalities can therefore help maintain reliable perception across a wider range of operating conditions.

This internship focused on improving the situational awareness of a Maritime Autonomous Surface Ship using land-based cameras. More specifically, the work investigated the combination of conventional RGB imagery with Long-Wave Infrared (LWIR) imagery. While RGB cameras capture information in the visible spectrum, LWIR cameras measure thermal radiation, providing complementary information about objects in the scene. The objective was to develop a perception pipeline capable of detecting and localizing boats using these two modalities.

1 Internship context

1.1 Presentation of the research group

The internship took place in the Technical University of Denmark (DTU), a technical university located in Lyngby, near Copenhagen. In DTU, I joined the Control, Robotics and Embodied AI (CREA) research group, which is part of the department of Electrical and Photonics Engineering, also called DTU Electro. The CREA research group develops intelligent robotic and automation systems by combining control theory, robotics, and artificial intelligence. It collaborates with academic and industrial partners to address real-world challenges in areas such as healthcare, energy, maritime systems, and sustainable industry.

1.2 Context of the project

The navigation of Maritime Autonomous Surface Ships (MASS) relies on onboard sensors to determine their localization and surroundings. In the case of remote control, an operator on land should be able to rely on trusted and reliable data. However, the data transmitted from the boat's sensors could be falsified or degraded. That is why it is important to develop land-based sensors the pilot can trust.

One type of land sensor used to improve situational awareness is cameras. Last year, ENSTA student Solène Boudot [1] worked on vision-based boat localization using RGB stereo cameras. However, RGB cameras lose reliability in harsh weather conditions (rain, fog, sun glare) or at night, since they depend on visible light. Long-Wave Infrared (LWIR) cameras, which rely on thermal radiation rather than visible light, remain effective in these degraded conditions and offer a complementary modality to RGB.

The objective of my internship was therefore to develop a pipeline combining LWIR and RGB imagery to detect and localize boats from the land-based platform, addressing the specific challenge of maintaining reliable object detection in conditions where RGB cameras alone are insufficient.

1.3 Presentation of the land platform



Figure 1: The land platform.

To address the problem of situational awareness from land, DTU developed a land-based platform to conduct data collection for research.

The land platform is composed of three cameras mounted on a rig: two RGB cameras at the extremities, separated by around 2 m, and one LWIR camera in the middle. It also has a W-band radar (millimeter-wave radar).

The data I worked with came from a DTU data campaign in Copenhagen Harbour in March 2026. The main boat, DANA IV (DTU's boat), starts relatively close to the camera (around 100 meters) and progressively moves away, up to 1400 meters. The weather was clear.

I also used data from the GNSS sensor on the DANA IV and AIS (Automatic Identification System, used for tracking ships) for ground truth of the ship's position.

2 Pipeline

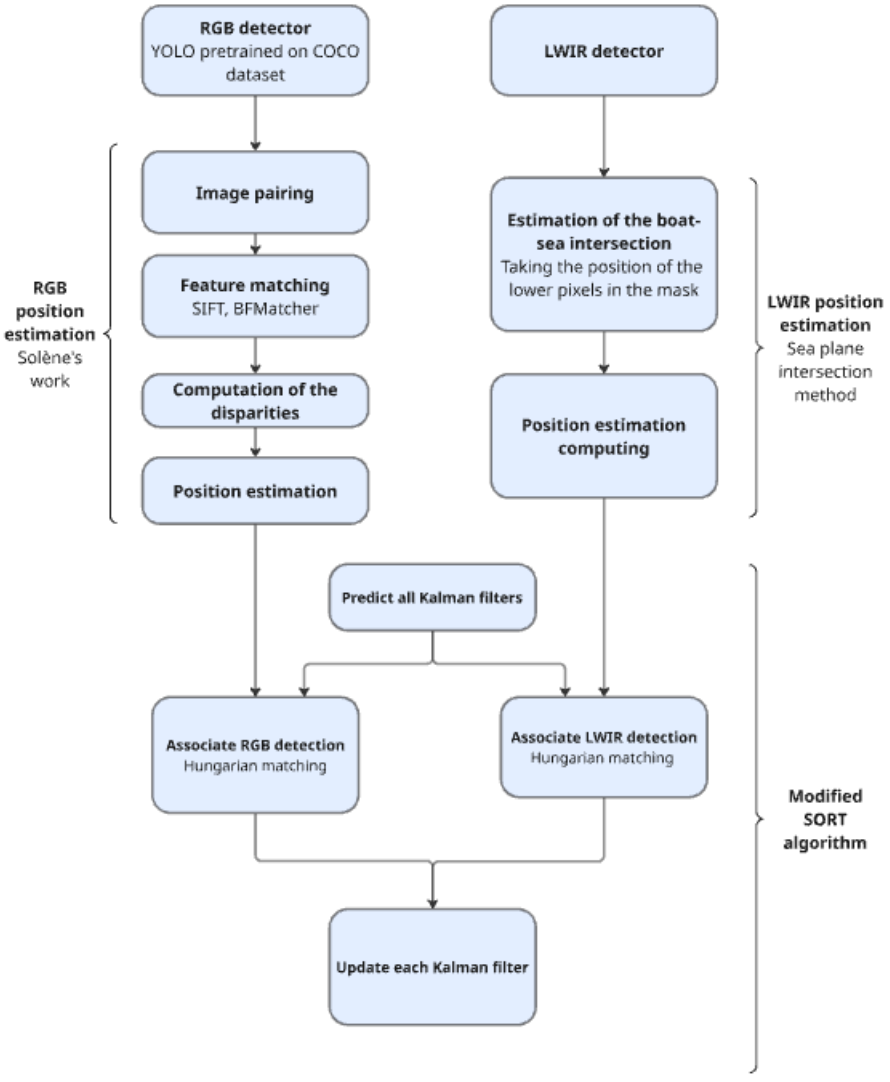


Figure 2: Overview of the proposed pipeline.

The pipeline has three parts : first we need to detect the boat, then estimate its position and then fuse the information from both modalities and track the boat.

2.1 Detection

Detection and position estimation within the RGB modality were carried out by Solène Boudot. Her approach begins with boat detection using YOLO, a pretrained neural network model. Position estimation is then performed using stereo vision. It consists of using two cameras to observe the same scene from slightly different viewpoints. This method is based on the parallax effect: because the cameras are separated by a known distance (the baseline), an object in the scene appears at a slightly different position in each image. This positional difference is called disparity. Given the disparity, the focal length of the cameras, and the baseline, we can estimate the depth of objects in the scene. Close objects produce a large disparity, while distant objects produce a small one. So given the images from the two aligned RGB cameras, she detects features on each boat using SIFT (Scale-Invariant Feature Transform) and matches them across the two images with a Brute-Force Matcher. The mean disparity of the matched features is then computed and used to estimate the boat's position [1].

Regarding the LWIR detector, since the sensor platform is fixed onshore, I first investigated background subtraction to detect boats in the LWIR images, using a Gaussian mixture model (OpenCV MOG2). Moving boats were detected very reliably, even boats far away. However, waves, boat wakes and sun glint produced persistent false detections, and slow or stationary boats were progressively absorbed into the background model. Finally, when a boat was close to the camera, its mask included a large part of its reflection on the water surface. Since the position estimation relies on the lower part of the mask, this reflection would bias the result.



Figure 3: Background subtraction (OpenCV MOG2) applied to an LWIR image. The mask includes the boat's reflection on the water, as well as small ripples.

Given these limitations, I turned to a deep learning approach. Deep learning detectors have previously been evaluated on maritime LWIR imagery [2]. I trained a YOLO11 segmentation model (YOLO, You Only Look Once) on the MassMIND maritime LWIR dataset to detect boats directly in the infrared spectrum [3] [4].

2.2 Position estimation

Since only one camera was available for the LWIR branch, stereo vision could not be used to estimate the position of the boats, and another method had to be found.

Several approaches exist for estimating the position of an object from a single camera:

— **Apparent size:** using the known real-world size of the object, together with the camera's focal length, to estimate distance from the size of its projection in the image. However, the real size of the boat is not necessarily known, so this method was not used here.

— **Position relative to the horizon:** using the position of the object with respect to the horizon line to estimate distance. This method could not be used here, since the images were captured in a harbour, where the horizon is not visible.
— **Sea-plane intersection:** using a known geometric relationship between the camera and a reference plane in the scene (here, the sea surface) to project image points onto that plane and recover their 3D position. This is the approach used in this project.

2.3 Fuse and track

The role of fusion is to reconcile detections coming from two independent detection branches: RGB stereo vision (Solène's work) and LWIR sea-plane intersection, so that the strengths of one modality can compensate for the weaknesses of the other (for instance, LWIR remaining reliable at night or in low visibility, when RGB may fail). Fusion of the two modalities can be performed at different levels : pixel level, detection level (bounding boxes) or measurement-level (real world position estimates).

Once a boat has been detected and localized at a given instant, its position must also be estimated continuously over time, even when detections are momentarily missing or noisy. This is the role of tracking : maintaining, for each boat, a coherent trajectory across successive frames, filtering out measurement noise, and predicting its position when a detection is temporarily unavailable.

An important design choice was therefore when to fuse the two modalities, and when to track the boats.

Fusing the two modalities first (for example, using a weighted average of the bounding box parameters) would require synchronization between the RGB and LWIR cameras, as well as accurate extrinsic calibration between them. I also find it more difficult to incorporate Solène's work, particularly the stereo-vision-based position estimation.

Tracking first would require two separate trackers, one for the RGB and one for the LWIR detection, thus increasing the computational cost of the pipeline. It would also introduce an additional track-to-track association problem to reconcile the two independent track histories — complexity that fusing first avoids.

Finally, I chose to perform tracking and fusion jointly, using a modified SORT algorithm. SORT stands for Simple Online and Realtime Tracking, and is commonly used to track detections from an object detector across frames [5].

A dedicated Kalman filter is maintained for each tracked boat. Whenever a new RGB or LWIR frame arrives, the newly detected boats are matched to the predicted Kalman filter states using the Hungarian algorithm. Instead of assigning detections to tracks one by one, the Hungarian algorithm looks at all detections and all predicted states together, and chooses the combination that minimizes the total distance (the sum of all match distances). Here, the distance between a detection and a track is the Euclidean distance in world coordinates.

Because RGB and LWIR frames can arrive asynchronously, each modality's detections would be matched independently against the same set of Kalman filters, allowing either camera to update a track's state whenever new data becomes available.

In practice, since only one main boat was present in the evaluation sequences and by lack of time, the Hungarian association step was not implemented: a single Kalman filter was maintained and updated directly using both RGB and LWIR detections. The matching step described above would however be needed to extend this approach to multiple boats.

3 Detection

To detect the boats, I used YOLO (You Only Look Once), a deep learning model for object detection. YOLO is a single-stage detector that uses a single neural network to jointly predict bounding boxes and classify objects,

making it suitable for real-time applications [6].

I first tested YOLO11 pretrained on the COCO dataset, a large-scale RGB object detection dataset including a boat category, directly on LWIR images. The model performed well overall, except for boats far away. I therefore fine-tuned YOLO on LWIR images, hoping to improve detection at longer distances.

YOLO11 segmentation was chosen over standard detection because it produces a pixel-level mask in addition to the bounding box. This mask gives a far more precise object boundary than a bounding box, which unavoidably includes background pixels, and this precision was expected to improve the accuracy of the position estimation. The availability of a pre-existing segmented LWIR dataset (MassMIND), which is rare for this modality, also supported this choice.

3.1 Dataset preparation

For training the YOLO model, I used the MassMIND dataset (Massachusetts Maritime INfrared Dataset) [4]. It contains over 2,900 maritime LWIR images. Each image is segmented and labeled with seven different classes : sky, water, bridge, obstacle, living obstacle, background, and self (the boat from which the images are taken). The obstacle class contains non living entities that are partially submerged or visible on the surface of water, for example ships, buoys, wooden pillars or other construction. The main objective is to detect objects belonging to this class. Although I could have chosen to train the model only on this class to obtain a lighter model, I decided to train it on all classes to reduce false positives, particularly to prevent background regions from being detected as obstacles.

For each LWIR image of the dataset, there is a semantically segmented annotated mask and an instance-segmented annotated mask. Semantic segmentation assigns a class label to each pixel in the image, whereas instance segmentation assigns a separate label to each object instance.



(a) LWIR Image (b) Semantic segmentation (c) Instance segmentation

To train the YOLO model with this dataset, I needed to convert the dataset masks into YOLO-format labels. The YOLO format represents each object instance in a text file containing its class and the coordinates of its segmentation polygon.

One issue I encountered was that YOLO segmentation labels represent each object instance using a single segmentation polygon. However, the dataset contains several cases where a single instance is represented by multiple polygons, for example when the sky or the water is occluded by a ship, or when a ship is partially occluded. One example is shown below.



Figure 5: Example of a boat instance and a background instance represented by multiple polygons.

To address this issue, I decided to merge the polygons belonging to the same instance for the obstacle and living obstacle class, as these polygons were generally located close to each other. For the water, bridge, sky, background and self classes, instances could sometimes be separated by large objects, such as boats. In these cases, I decided not to merge the polygons, as doing so could introduce unwanted information into the annotations, particularly for the detection of boats. Moreover, for these classes, it is not important for the purpose of this work whether each separate polygon is treated as a distinct instance.

Finally, I split the dataset into training images (70%), validation images (15%), and test images (15 %). The validation set is used during training to monitor the model's performance and detect overfitting (when the model performs well on the training data, but fails to generalize to new, unseen data) while the test set is used to evaluate the model's performance after training. I then verified that the splits were consistent in terms of class distribution and number of instances.

3.2 Training

To retrain the YOLO11 model on LWIR images, we used weights pretrained on the COCO dataset. Training from scratch, i.e. starting from random weight values, would be computationally expensive and would require a very large dataset for the model to learn object features. Since object features are largely similar across the RGB and LWIR domains, transfer learning offers a more efficient alternative. As the LWIR images are grayscale, however, we have to replace the first layer to accept 1 channel instead of 3. The weights of this new layer are randomly initialized.

I chose not to apply preprocessing, like denoizing or contrast-enhancement, following the reasoning that deep learning models can often learn effectively from raw, low-contrast LWIR data without explicit correction, and to keep the pipeline lightweight.

Tuning the model's training parameters was a bit difficult. I worked iteratively : setting parameters that seemed reasonable, training the model, then visualizing the loss curves and the metrics to check how it performed and adjusting from there.

The model was trained using GPUs provided by DTU. For the best-performing model, I set the number of epochs to 500, but the best model occurred around epoch 180. Ultralytics (the framework used to train YOLO) automatically saves the weights of the best-performing checkpoint during training [7]. I set the batch size to 7 to avoid GPU out-of-memory errors. Finally, no layers were frozen during training.

My goal was to detect the DANA IV for as long as possible as it moves further away. However, detecting small objects is a known challenge in computer vision: as objects occupy fewer pixels, their characteristic features

become harder to detect. One solution is to increase the training image size, so that objects occupy more pixels [2]. YOLO's default training image size is 640 pixels; I increased it to 960 and 1280 pixels. The 960px model performed well, but I expected the 1280px model to perform even better. Instead, it produced a lot more false detections in the background, and detected the boat at a much shorter distance than the 960px model. I didn't identify a clear underlying cause.

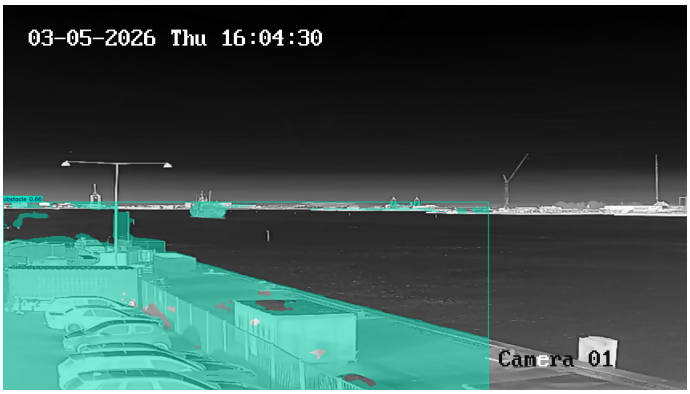


Figure 6: Example of an image where the boat is not detected, with a false boat detection in the background, using the model trained at 1280px.

Data augmentation:

Data augmentation is a technique used to artificially increase the size and diversity of a training dataset by applying transformations to existing images. The following augmentations were used:

— Mosaic augmentation: set to 0.5, combining four training images into a single image to simulate more complex scenes and object configurations.

— Hue and saturation: disabled (set to 0), since LWIR images are single-channel.

— Value: was set to 0.3 to simulate variations in brightness.

— Copy-paste: set to 0.1, allowing objects from one image to be randomly pasted into another to increase scene diversity.

[7]

3.3 Evaluation

The retrained LWIR model detected the boat up to 1190m, outperforming the COCO-pretrained model on the same LWIR images (1080m) — showing the value of domain-specific retraining. Interestingly, I was surprised by how well the COCO-pretrained model performed on LWIR images, given it was never trained on thermal data. However, it still fell short of the COCO-pretrained model's performance on RGB images (1250m), reflecting the extra difficulty of small-object detection in thermal imagery compared to RGB, as well as COCO's much larger and more diverse training set.

Model	Training modality	Inference modality	Max detection distance
YOLO11-seg (retrained on MassMIND)	LWIR	LWIR	1190 m
YOLO (COCO-pretrained, not retrained)	RGB	LWIR	1080 m
YOLO (COCO-pretrained, not retrained)	RGB	RGB	1250 m

Table 1 – Maximum detection distance of the DANA IV boat by model, training modality, and inference modality.

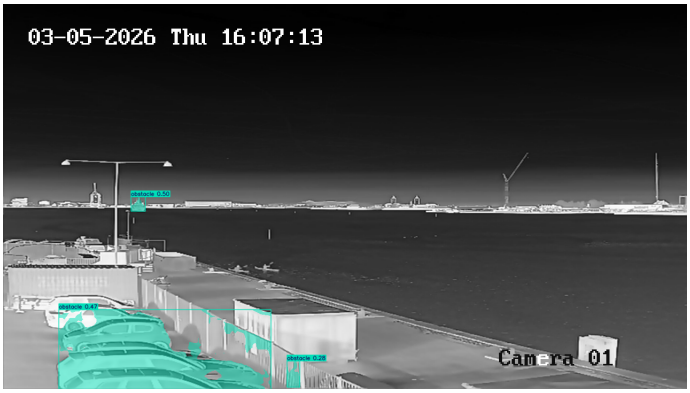
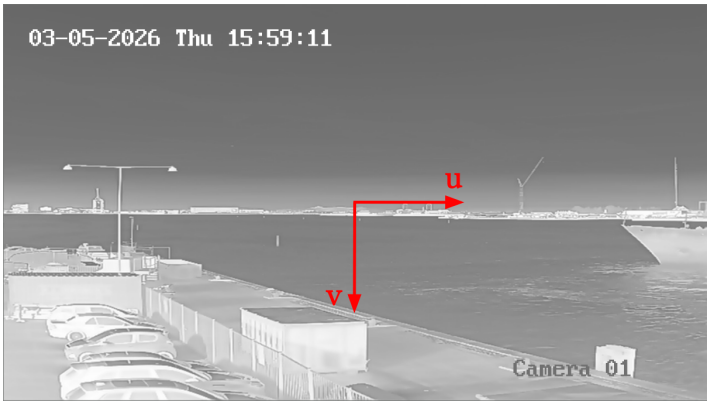


Figure 7: Last detection of the DANA IV at 1190m by the retrained model.

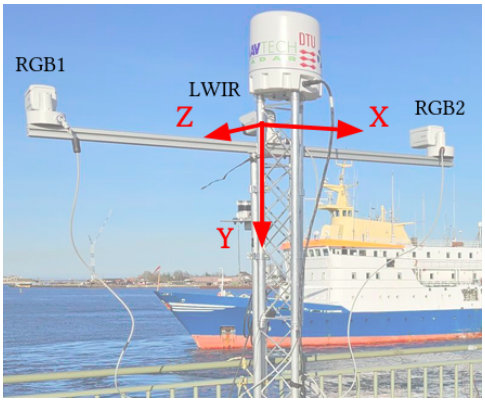
In the figure above, the YOLO model retrained on MassMIND detects the boat even when it occupies a small part of the image and partly merges with the background. However, it fails to detect the kayaks, which is likely due to the absence of kayaks in the training dataset. The model also detects part of the background as a boat. Since only one boat is tracked, only the detection with the highest confidence score is kept in each frame, which limits the impact of these false detections. This remains a simplification: a background detection with a higher score than the boat would be selected instead, and with several boats, false detections would have a greater impact since they could generate false tracks.

Improving the retrained model's performance would require a better dataset, with more variety in boat types and environments, and a larger number of images. Unfortunately, the availability of LWIR maritime datasets is limited [8].

4 Position estimation



(a) Image frame



(b) World frame

The pinhole camera model describes the projection of a 3D point in the camera frame onto the 2D image plane.

The intrinsic matrix K maps the 3D coordinates of a point expressed in the camera frame to its pixel coordinates in the image:

$$\lambda \cdot \begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = K \cdot \begin{bmatrix} X_{\text{cam}} \\ Y_{\text{cam}} \\ Z_{\text{cam}} \end{bmatrix} = \begin{bmatrix} f_x & 0 & c_x \\ 0 & f_y & c_y \\ 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} X_{\text{cam}} \\ Y_{\text{cam}} \\ Z_{\text{cam}} \end{bmatrix} \tag{1}$$

where (u, v) are the pixel coordinates, $(X_{\text{cam}}, Y_{\text{cam}}, Z_{\text{cam}})$ are the 3D coordinates in the camera frame, f_x and f_y are the focal lengths expressed in pixels, and (c_x, c_y) is the principal point. The scale factor λ corresponds to the depth of the point in the camera frame, such that $\lambda = Z_{\text{cam}}$ under the standard pinhole camera convention.

The extrinsic parameters, represented by a rotation matrix R and a translation vector t, define the transformation from the world frame to the camera frame:

$$\begin{bmatrix} X_{\text{cam}} \\ Y_{\text{cam}} \\ Z_{\text{cam}} \end{bmatrix} = R \cdot \begin{bmatrix} X \\ Y \\ Z \end{bmatrix} + t \tag{2}$$

Combining the intrinsic and extrinsic parameters gives the complete projection model:

$$\lambda \begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = K \begin{bmatrix} R & t \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix}. \tag{3}$$

4.1 Calibration

The calibration of a camera consists of estimating its intrinsic and extrinsic parameters.

Here my main objective was to estimate the focal length. The standard way to calibrate a camera is with a checkerboard pattern. For an LWIR camera, this requires a checkerboard with sufficient thermal contrast for the pattern to be visible in the LWIR spectrum. I did not have access to such a checkerboard, and although I could have built one myself, the camera may have moved since the campaign, so any calibration performed at a later date would not necessarily reflect its position during data acquisition.

I therefore tried to calibrate the LWIR camera using two different methods. First, I manually mapped corresponding points across the three images captured by the three cameras. Since the RGB cameras were already calibrated, I used triangulation to compute the 3D coordinates of the points in the coordinate frame of the left camera.

This method worked well, but it meant that the LWIR camera calibration relied on both the triangulation results and the existing calibration of the RGB cameras, neither of which is perfect.

The second method used the GPS coordinates of the boat. I preferred this approach because the GPS coordinates were more accurate than the 3D coordinates obtained through triangulation.

For this method, I selected a point on the boat and tracked it over 36 frames. I then matched these 2D image coordinates with the 3D GPS coordinates of the boat at the same timestamps. To do this, I transformed the GPS coordinates, given in latitude and longitude, into coordinates in the camera frame using the known position and orientation of the camera. Having both the 2D image coordinates and the corresponding 3D object coordinates, I used OpenCV's `calibrateCameraExtended` function to calibrate the LWIR camera.



Figure 9: The point tracked over 36 frames.

The other important parameter to know is the vertical coordinate of the principal point of the camera c_y. This is the value against which v is compared in the sea-plane intersection method. If v equals c_y, the boat lies on the horizon; conversely, the further v is from c_y, the closer the boat is to the camera.

To have a precise evaluation of c_y I selected a pixel at the intersection of the line of buildings and the sea. I then used Google Maps to obtain the world coordinates of this point and computed its distance to the camera. Finally, I inverted the sea-plane intersection formula to estimate c_y.

$$c_y = v - \frac{f \cdot h}{d}$$

with v (px) the vertical image coordinate of the selected point, d the distance (m) of the point to the camera, f the previously estimated focal length and h the known height of the camera above sea level.

For the estimation of the extrinsic parameters, I used the measured real-world distance as the translation and assumed an identity rotation matrix, $R = I$. The extrinsic parameters obtained with OpenCV were not consistent with the actual camera configuration and were therefore not used.

4.2 Sea-plane intersection

To estimate the position of detected vessels using only an LWIR camera, the sea plane intersection method was employed. Given the pixel coordinates (u, v) of a detection, we want to recover its position (X, Y, Z) in the world frame. However, a single 2D point is insufficient to determine a unique 3D position, it only constrains the solution to a ray originating from the camera center. We need to add an additional constraint. To do so we assume that all detected objects lie on the sea surface, modelled as a flat horizontal plane at $Y = h$. h is the height of the camera with respect to the sea level. We also assume the camera has no rotation with respect to the world frame ($R = I$): the tilt of the camera is not measured. To a first approximation, estimating c_y from a point at a known distance absorbs the effect of a small constant tilt.

For each detection, the contact point with the sea surface is taken at the bottom of the mask, with its vertical coordinate v averaged over the lower boundary of the mask (Section 4.3). Knowing the camera height above the sea plane, the ray through this point is then intersected with the $Y = h$ plane to recover a unique world position (X, h, Z).

Theory :

By expanding the intrinsic matrix equation (1), the pixel coordinates can be expressed as :

$$u = f_x \cdot \frac{X_{\text{cam}}}{Z_{\text{cam}}} + c_x \qquad v = f_y \cdot \frac{Y_{\text{cam}}}{Z_{\text{cam}}} + c_y$$

Conversely, a pixel (u, v) defines a ray in the camera frame. Setting $Z_{\text{cam}} = 1$ gives the corresponding ray direction:

$$d_{cam} = \begin{bmatrix} \frac{u-c_x}{f_x} \\ \frac{v-c_y}{f_y} \\ 1 \end{bmatrix}$$

Since R defines the transformation from the world frame to the camera frame, the ray direction expressed in the world frame is:

$$\mathbf{d}_{\text{world}} = R^\top \cdot \mathbf{d}_{\text{cam}}$$

The coordinates of a point P in the camera frame are given by :

$$P_{cam} = RP_{world} + t$$

The camera center C in the world frame is obtained by setting $P_{\text{cam}} = \mathbf{0}$:

$$\mathbf{0} = RC + t,$$

which gives :

$$C = -R^\top \cdot \mathbf{t}$$

The 3D ray corresponding to pixel (u, v) can therefore be expressed as:

$$P(\lambda) = C + \lambda \cdot \mathbf{d}_{\text{world}}, \quad \lambda > 0$$

with

$$P = \begin{bmatrix} X \\ Y \\ Z \end{bmatrix}$$

Assuming that the sea surface is represented by the plane $Y = h$, substituting the ray equation into the plane equation gives:

$$Y = C_Y + \lambda d_{Y,world} = h$$

And therefore :

$$\lambda = \frac{-C_Y + h}{d_{Y,world}}$$

The corresponding coordinates on the sea surface are then:

$$X = C_X + \lambda \cdot d_{X,world}$$

$$Z = C_Z + \lambda \cdot d_{Z,world}$$

Under the assumptions $R = I$, $C = [0, 0, 0]^\top$, and $f_x = f_y = f$, this simplifies to:

$$Z = \frac{\mathrm{h} \cdot f}{v - c_y} \qquad X = \frac{\mathrm{h} \cdot (u - c_x)}{v - c_y}$$

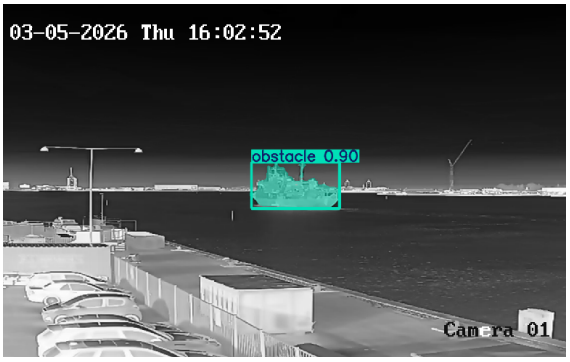
where f is the focal length, (c_x, c_y) is the principal point of the camera, and h is the known height of the camera above the sea surface.

4.3 Determining the waterline pixel coordinates

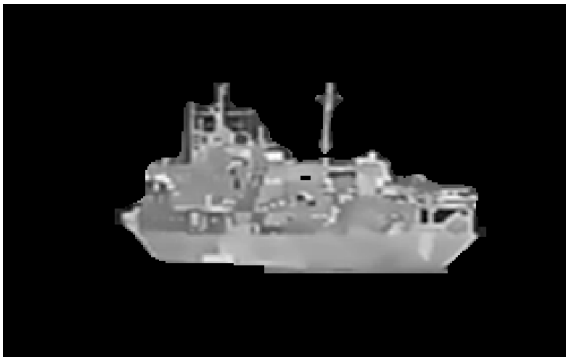
To determine the intersection between the boat and the sea in the image, I first used the v-coordinate of the lowest pixel of the mask (the image v-axis points downward, so this is the maximum v of the mask). However, I later learned that the YOLO segmentation mask has a precision of about 4 pixels, which produced a stepped distance plot rather than a smooth one. Moreover, any noise that extends the mask downward biases this pixel.

To improve the precision of v, I averaged over the lower pixels of the mask. Specifically, I selected the columns corresponding to the middle 50 % of the boat's width, discarding the outer 25 % on each side (where bow and stern tapering or noise is more likely to cause an inconsistent bottom edge). For each column, I took the maximum v among the mask pixels in that column (i.e. the lowest point of the mask at that u position) and then computed the average of these values across columns.

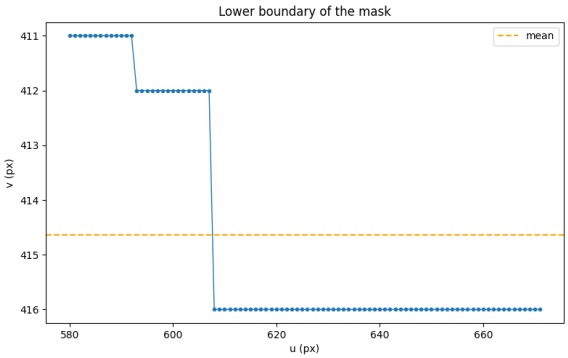
This produced a smoother and more precise v estimate, since relying on a single maximum-v pixel introduced instability in the distance calculation.



(a) Detection of the boat.



(b) Mask of the boat.



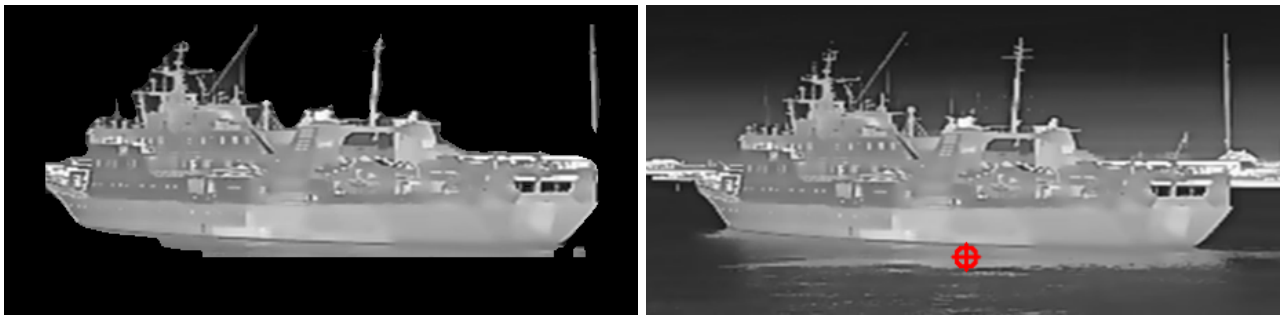
(c) Calculating the average v of the lower boundary of the mask.



(d) Resulting sea-boat intersection point (in red).

Figure 10: Waterline determination example.

One challenge of this method is dealing with the thermal reflection of the boat in the sea. Indeed, warm objects like boats emit heat that reflects on the water's surface similarly to light, which makes it harder to detect the true bottom edge of the boat. Fortunately, the mask often included only a small part of the reflection, and averaging the v-values also helped reduce this problem to some extent. One possible solution to address this issue would be to use an edge-detection technique based on the vertical intensity gradient.



(a) Example mask including part of the boat's reflection. (b) Resulting sea-boat intersection point (in red).

4.4 Uncertainty

Uncertainty propagation was used to compute the uncertainty of the calculated distance. I estimated the uncertainty on the detection of the sea-plane intersection $\boldsymbol{v}$ to 4 pixels but it depends a lot if there is a reflection or if the boat is far away. The uncertainty on the focal length f is given by OpenCV and is estimated to 350 pixels.

Recall of the sea-plane intersection formula :

$$z = \frac{hf}{v - c_y}$$

The uncertainty on the denominator is :

$$u_{v-c_y} = \sqrt{u_v^2 + u_{c_y}^2}$$

Finally the uncertainty on the distance z in meters is :

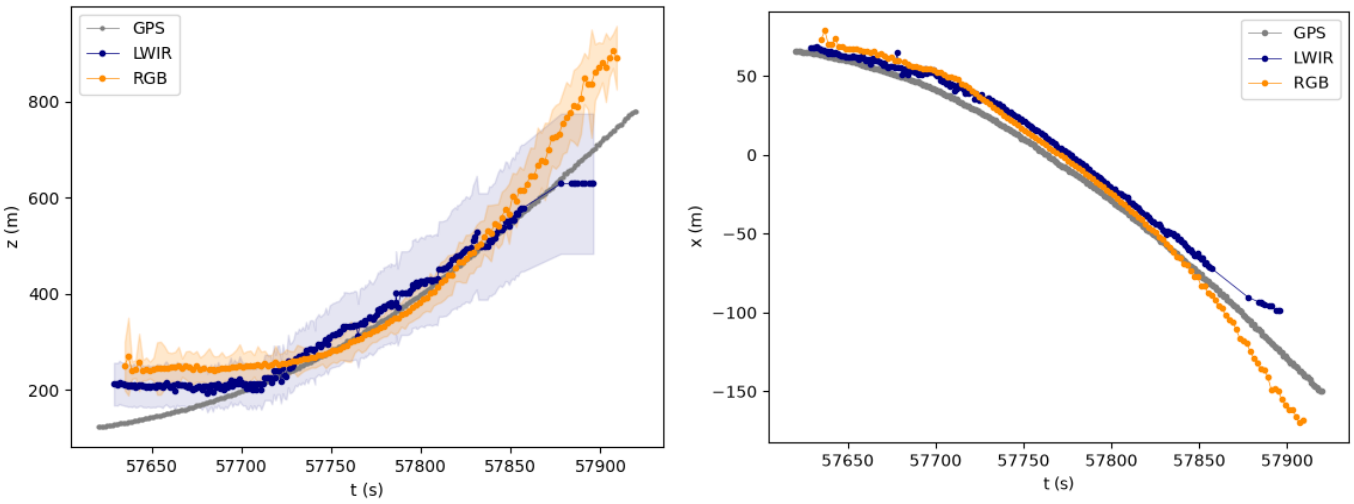
$$u_z = z\sqrt{\left(\frac{u_h}{h}\right)^2 + \left(\frac{u_f}{f}\right)^2 + \left(\frac{u_{v-c_y}}{v - c_y}\right)^2}$$

To give a rough idea, this uncertainty is around 44m when the boat is at a distance z=210m and around 250m when the boat is at a z=800m. This uncertainty is quite significant.

Here are the terms :
$\frac{u_h}{h} = 0.06$ $\frac{u_f}{f} = 0.19$ at 210m : $\frac{u_{v-c_y}}{v-c_y} = 0.06$ at 800m : $\frac{u_{v-c_y}}{v-c_y} = 0.24$

A better calibration would help significantly at low and medium range, since the uncertainty on the focal length f is the dominant term there. At long range, the uncertainty on v becomes the largest contributor, so improving the sea-plane intersection detection — for instance with a higher-resolution LWIR sensor — would likely bring the greatest gain.

4.5 Results



(a) Trajectory on the z-axis with respect to time

(b) Trajectory on the x-axis with respect to time

Figure 12: Results for the boat's localization.

The GPS trajectory, shown in gray, is used as the ground truth.

For the Z coordinate, both the LWIR- and RGB-based position estimates overestimate the distance of the boat before $t = 57700$ s. A possible explanation is that, during this period, only part of the boat is visible in the image, which affects the localization of the detected bounding box. Between $t = 57700$ s and $t = 57850$ s, both modalities provide accurate estimates, closely matching the GPS trajectory. After $t = 57850$ s, the RGB estimates progressively overestimate the distance.

Overall, the boat is accurately localized up to an average distance of approximately 550 m from the camera. Beyond this range, the estimation uncertainty increases significantly, as illustrated by the growing confidence intervals.

At the end of the internship, it was found that the baseline used for the RGB stereo-vision system was inaccurate. This may explain part of the observed error in the RGB estimates, particularly at larger distances. Since stereo depth is given by

$$Z = \frac{fB}{d},$$

where B is the stereo baseline, any error in the baseline directly introduces a proportional error in the estimated depth.

5 Kalman filter

5.1 Theory

I implemented a Kalman filter to fuse the position estimates from RGB stereo vision and from the LWIR sea-plane intersection. It predicts the state with a motion model and corrects it with the position measurements that arrive (from RGB or LWIR). This also filters out sensor noise in the process. In each update, the estimated state is a weighted average of the predicted state and the new measurement, where the weight (the Kalman gain) is determined by the uncertainty of the process and the measurement — the source with lower uncertainty

is trusted more [9].

The system is the boat I'm tracking and its state $\mathbf{x}$ is :

$$\mathbf{x} = \begin{bmatrix} x \\ z \\ \dot{x} \\ \dot{z} \end{bmatrix}$$

where x and z are the world coordinates of the boat in the left camera frame in meters, $\dot{x}$ and $\dot{z}$ are the velocities.

Prediction step :

We model the boat's motion using a constant velocity model, which assumes that there is no acceleration between consecutive time steps [9]. Given a time step Δt the state transition model is defined as:

$$\begin{aligned} x_k &= x_{k-1} + \Delta t \dot{x}_{k-1} \\ z_k &= z_{k-1} + \Delta t \dot{z}_{k-1} \\ \dot{x}_k &= \dot{x}_{k-1} \\ \dot{z}_k &= \dot{z}_{k-1} \end{aligned}$$

Written in matrix form :

$$\bar{\mathbf{x}} = \mathbf{F}\mathbf{x}$$

$$F = \begin{bmatrix} 1 & 0 & \Delta t & 0 \\ 0 & 1 & 0 & \Delta t \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Where $\bar{\mathbf{x}}$ is the prior state and $\mathbf{F}$ is the state transition matrix.
The prior covariance ($\mathbf{P}$ after prediction) $\bar{\mathbf{P}}$ is computed as :

$$\bar{\mathbf{P}} = \mathbf{F}\mathbf{P}\mathbf{F}^{\mathbf{T}} + \mathbf{Q}$$

$\bar{\mathbf{P}}$ represents the uncertainty of the predicted state, combining the propagated uncertainty from the previous estimate with the added process noise. $\mathbf{Q}$ is the process noise covariance matrix, modeling the uncertainty in the constant-velocity assumption between time steps.

To find the form of $\mathbf{Q}$ we use the discrete white noise acceleration model. It assumes the acceleration is constant during each time step Δt but differs by a random white noise increment for each time period :

$$f(x) = Fx + \Gamma w$$

where Γ is the noise gain of the system and w the constant piecewise acceleration.

$$w(t) = \begin{bmatrix} w_x(t) \\ w_z(t) \end{bmatrix}$$

In one time period the change in velocity will be $w(t)\Delta t$ and the change in position $\frac{w(t)\Delta t^2}{2}$.

$$\Gamma = \begin{bmatrix} \frac{\Delta t^2}{2} & 0 \\ 0 & \frac{\Delta t^2}{2} \\ \Delta t & 0 \\ 0 & \Delta t \end{bmatrix}$$

By definition :

$$Q = COV(\Gamma w(t)) = \Gamma \sigma_a^2 \Gamma^T$$

Finally :

$$Q = \sigma_a^2 \begin{bmatrix} \frac{\Delta t^4}{4} & 0 & \frac{\Delta t^3}{2} & 0 \\ 0 & \frac{\Delta t^4}{4} & 0 & \frac{\Delta t^3}{2} \\ \frac{\Delta t^3}{2} & 0 & \Delta t^2 & 0 \\ 0 & \frac{\Delta t^3}{2} & 0 & \Delta t^2 \end{bmatrix}$$

where σ_a^2 is the acceleration variance.
We choose

$$\frac{1}{2}\Delta a < \sigma_a < \Delta a$$

where Δa is the maximum acceleration change between sample periods.

Here σ_a is set to $0.5 m.s^{-2}$.

Update Step :

Every time we made a measurement (with the LWIR or RGB camera) we update the state according to the following equations :

$$\mathbf{y} = \mathbf{z} - \mathbf{H}\bar{\mathbf{x}}$$
$$\mathbf{K} = \bar{\mathbf{P}}\mathbf{H}^{\mathsf{T}}(\mathbf{H}\bar{\mathbf{P}}\mathbf{H}^{\mathsf{T}} + \mathbf{R})^{-1}$$
$$\mathbf{x} = \bar{\mathbf{x}} + \mathbf{K}\mathbf{y}$$
$$\mathbf{P} = (\mathbf{I} - \mathbf{K}\mathbf{H})\bar{\mathbf{P}}$$

With y the residual, K the Kalman gain, R the measurement noise covariance and H is the measurement function.

Here H is :

$$H = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix}$$

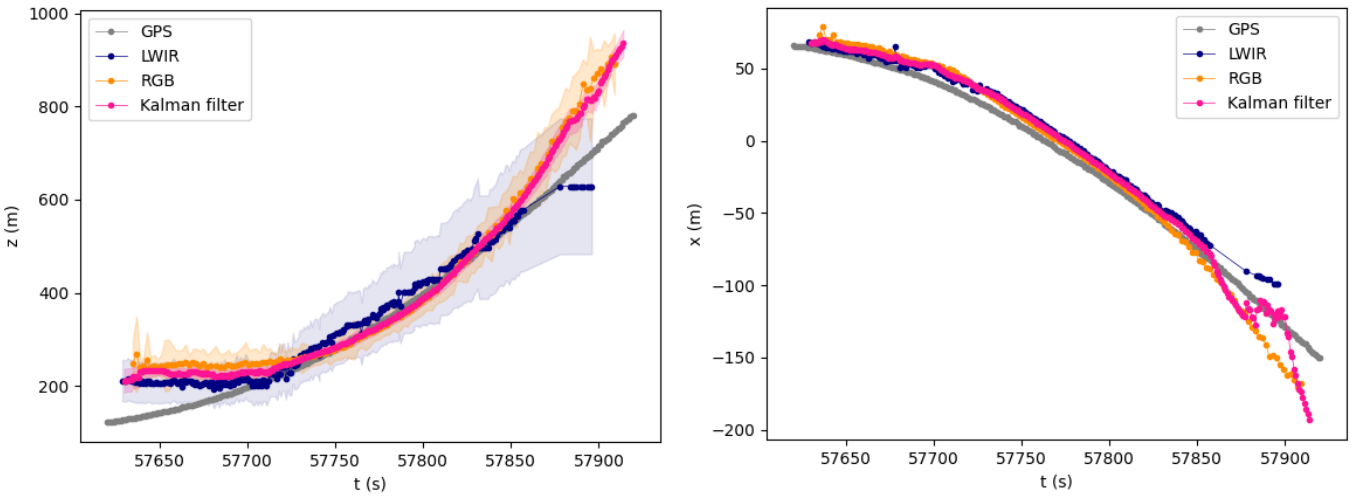
since we only measure x and z.

$$R = \begin{bmatrix} \sigma_{x_x}^2 & 0 \\ 0 & \sigma_{z_z}^2 \end{bmatrix}$$

The covariance between x and z is neglected, so the measurement noise covariance matrix R is diagonal. For the z coordinate, the uncertainty is adaptive: it depends on whether the measurement comes from the LWIR or the RGB modality, and it is larger when the boat is far away. For the LWIR branch, it is computed by propagating the uncertainties on the calibration and on the detection of the sea-plane intersection (Section 4.4). For the RGB branch, it is the standard deviation of the distances computed from the disparities.

For the x coordinate, the uncertainty is set to a constant value of 10 m for both modalities, chosen empirically from the observed difference with the GPS. This is a simplification, since the uncertainty on x should in theory also grow with the distance to the boat.

5.2 Results



(a) Trajectory on the z-axis with respect to time

(b) Trajectory on the x-axis with respect to time

Figure 13: Results for the boat's localization and the Kalman filter.

The Kalman filter effectively smooths the position estimates while combining information from both modalities. For the Z coordinate, the filtered estimate follows the RGB position estimate more closely, as the RGB branch has a lower estimated uncertainty than the LWIR branch. In addition, the Kalman filter's prediction step makes it possible to bridge detection gaps, for instance when the LWIR branch temporarily fails to detect the vessel: the state is propagated using the motion model, and the track continues until a new measurement becomes available to correct it.

Conclusion

This work demonstrated that boats can be detected with promising performance in long-wave infrared (LWIR) images and localised in the world frame using a single thermal camera. Boats are first detected and segmented in the LWIR image with a YOLO11-seg model retrained on the MassMIND dataset. The pixel at the waterline of the boat is then converted into a viewing ray using the camera intrinsics and extrinsics, and this ray is intersected with the sea plane, assumed to be at a known distance below the camera, to recover the boat's position. A Kalman filter was then implemented to fuse the LWIR and RGB position estimates and track the boat over time. The pipeline could be extended to handle several boats simultaneously, with detections matched to their tracks using Hungarian algorithm.

On the DANA IV sequence, the proposed approach successfully estimated the boat's position in the world frame. The localisation became less stable at greater distances, highlighting the sensitivity of the geometric estimation to camera height and calibration uncertainties. Combined with the RGB stereo branch, this LWIR-based localisation should provide a complementary measurement in low-light and adverse conditions.

Several improvements are possible. The LWIR intrinsic calibration (focal length and optical centre) and the extrinsic parameters (camera tilt and height above sea level) should be known precisely, as they are the main sources of position error. Detection could be improved with a more diverse training dataset and better-tuned hyperparameters. Multi-boat tracking should be tested on sequences with several boats, and the whole pipeline should be evaluated on other datasets and in adverse conditions such as fog. Finally, real-time operation should be tested, and the computation time and hardware requirements of the pipeline should be measured.

This work will be reused in future research at DTU. In particular, Morgan Louédec will use it to track an Otter, and his work on factor graphs for detecting sensor failures will benefit from an additional, independent position measurement.

This internship provided me with the opportunity to explore and strengthen my knowledge in various fields, including machine learning, computer vision, Kalman filtering, 3D vision, and experimental design. It allowed me to apply these concepts to a real-world maritime perception problem and provided valuable experience in developing, evaluating, and analysing a complete perception pipeline.

References

[1] Solène Boudot. *Vision-Based Boat Localization Using Stereo Cameras for Remote Situational Awareness*. Internship Report. Danmarks Tekniske Universitet, 2025.

[2] Frederik Emil Thorsson Schöller et al. « Assessing Deep-learning Methods for Object Detection at Sea from LWIR Images ». In: *IFAC-PapersOnLine* 52.21 (2019), pp. 64–71. DOI: 10.1016/j.ifacol.2019.12.284.

[3] Glenn Jocher and Jing Qiu. *Ultralytics YOLO11*. Version 11.0.0. 2024. URL: https://github.com/ultralytics/ultralytics.

[4] Shailesh Nirgudkar et al. *MassMIND: Massachusetts Marine INfrared Dataset*. Open access. DOI: https://doi.org/10.1177/02783649231153020.

[5] Alex Bewley et al. « Simple Online and Realtime Tracking ». In: *2016 IEEE International Conference on Image Processing (ICIP)*. 2016, pp. 3464–3468. DOI: 10.1109/ICIP.2016.7533003.

[6] Joseph Redmon et al. *You Only Look Once: Unified, Real-Time Object Detection*. May 9, 2016. DOI: 10.48550/arXiv.1506.02640. arXiv: 1506.02640[cs.CV]. URL: http://arxiv.org/abs/1506.02640 (visited on 09/30/2026).

[7] Ultralytics. *Train - Ultralytics YOLO Docs*. URL: https://docs.ultralytics.com/modes/train (visited on 09/30/2026).

[8] Hongguang Lyu et al. « Sea-Surface Object Detection Based on Electro-Optical Sensors: A Review ». In: *IEEE Intelligent Transportation Systems Magazine* (2023), pp. 190–216. DOI: 10.1109/MITS.2022.3198334.

[9] Roger R. Labbe. *Kalman and Bayesian Filters in Python*. https://github.com/rlabbe/Kalman-and-Bayesian-Filters-in-Python. Jupyter Notebook. 2015.