Smart Structures and Systems, *Vol. 36, No. 2 (2025) 71-82*
https://doi.org/10.12989/sss.2025.36.2.071

Identification of ship trajectory using deep learning-based segmentation and stereovision

Hai-Wei Wang and Rih-Teng Wu*

Department of Civil Engineering, National Taiwan University, Taipei, Taiwan

(*Received May 24, 2024, Revised December 23, 2024, Accepted February 19, 2025*)

Abstract. River transportation is a significant component of the overall transportation system. Typically, there are surveillance cameras implemented on river bank to avoid collisions between ships and bridges across rivers. However, some of the routes may only contain limited or malfunctioned cameras, making the monitoring of ships occluded. In this study, we propose a deep learning-based framework that identifies the trajectory of a ship in the real world by using the surveillance videos. The proposed framework consists of three modules: object detection, object tracking, and coordinate projection. We implement the Mask R-CNN model for object detection to determine the ship position in each video frame and compute the ship centroid as the image coordinates of the ship. We then employ DeepSort as the object tracker, which matches and tracks the detected object in each frame and combines all instances of object detection in the video to output the ship trajectory. For coordinate projection, we incorporate the P3P method and Zhang's algorithm to determine the intrinsic matrix and extrinsic matrix, respectively. The image coordinates of the ships are therefore converted into world coordinates. In addition, we develop an approach to calibrate the ship trajectory out of the coverage using the results from multi-camera triangulation. Meanwhile, the continuity in ship trajectory is enhanced as well. Results demonstrate that the ship trajectory becomes smoother in the evaluation using acceleration variability and directional change. The proposed approach reduces the acceleration variability score from 2.75 to 1.54 and improves the directional change score from 0.85 to 0.09.

Keywords: coordinate projection; deep learning; instance segmentation; object tracking; ship trajectory identification; triangulation

1. Introduction

1.1 Motivation

River transportation is a significant component of the overall transportation system. However, bridges across rivers often cause collisions because of their low navigation clearance and the associated poor navigation conditions. Therefore, some design requirements for bridge collision avoidance are needed (Campbell *et al*. 2012, Chauvin *et al*. 2013, Polvara *et al*. 2018, Zhao *et al*. 2023, Wang *et al*. 2019, 2023). In recent years, vision-based object recognition methods have been developed rapidly and used widely. For example, in object tracking. Instance segmentation models can identify targets in each frame of a video and predict the trajectories of these targets by using the information of the previous and subsequent frames. Then, real-world object tracking is realized using coordinate conversion algorithms. In this report, we propose a framework based on a computer vision mechanism such as instance segmentation or triangulation. It can determine the real-world track of a specified ship by processing videos of it captured from a camera by the river bank, camera calibration coefficient. This framework can be used as a bridge collision avoidance system.

1.2 Related work

1.2.1 Object detection

Girshick *et al*. (2014) proposed a deep learning-based object detection model called region-based convolutional neural network (R-CNN) that predicts objects by using a bounding box. It comprises three modules: creation of region proposals by means of selective search, feature extraction by using AlexNet (Krizhevsky *et al*. 2012), and classification of region proposals by using support vector machine (SVM). R-CNN tightens the bounding box by means of linear regression to improve its intersection over union (IoU) value. On the VOC 2010 test, the average detection precision (%) of R-CNN was 53.7% (Everingham *et al*. 2010). Because of the success of R-CNN, various variants have been developed to improve its performance. Girshick modified a few defects of R-CNN and proposed a new model called Fast R-CNN (Girshick 2015).

In Fast R-CNN, a new mechanism called region of interest pooling (RoIPooling) was introduced, and a CNN model, classifier, and linear regression were combined into a network to increase the speed of the model. Ren *et al*. (2015) proposed a more progressive R-CNN model called Faster R-CNN. In this model, they introduced a new mechanism called region proposal network (RPN) that greatly reduces the time required by the model to generate

*Corresponding author, Assistant Professor,
E-mail: rihtengwu@ntu.edu.tw

ISSN: 1738-1584 (Print), 1738-1991 (Online)

region proposals. He *et al*. (2017) proposed an instance segmentation model called Mask R-CNN. This model predicts an object by using not only a bounding box but also a mask to improve the position information to the pixel level. A fully convolutional network (FCN) was added on top of the CNN feature of Faster R-CNN to generate a mask (segment output). The FCN works by adding a branch to the Faster R-CNN network to output a binary mask that states whether a given pixel is a part of an object. In addition, Girshick proposed the RoIAlign method to adjust the RoIPooling method and achieve more accurate alignment.

1.2.2 Object tracking and projection

Bewley *et al*. (2016) proposed an object tracker called SORT that can match each detection and track in a video. It uses the Kalman filter and Hungarian algorithm to obtain motion information and object relationship information by using the size and position of the bounding box. SORT has a fast calculation speed. However, because the model only matches the track of the previous frame with that of the current frame, the target is recognized as another track because of short-term occlusion. Wojke *et al*. (2017) proposed a more progressive SORT algorism called DeepSort that overcame the short-term occlusion issue. DeepSort uses not only information pertaining to the size and position of the bounding box but also a neural net to obtain appearance information. Although this method is marginally slower, it can address short-term occlusions more effectively.

Triangulation (Hartley and Sturm 1997) serves as a fundamental geometric principle with diverse applications across fields ranging from surveying and navigation to computer vision and robotics. At its core, triangulation involves the determination of an unknown point location by measuring angles from known points or observing its projections from multiple viewpoints. This technique plays a crucial role in computer vision tasks such as 3D reconstruction, object localization, and camera pose estimation. By analyzing the projections of points from multiple camera viewpoints, triangulation algorithms can reconstruct the underlying 3D structure of a scene. Some application including ship trajectory identification implement coordinate projection using this method (Avidan and Shashua 2000).

1.2.3 Deep learning-based ship identification

In recent years, with the rise of deep learning, there have been revolutionary advances in ship identification based on images (Zwemer *et al*. 2018, Štepec *et al*. 2019, Kartal and Duman 2019, Stofa *et al*. 2020, Gupta *et al*. 2021, Yildirim and Kavzoglu 2021, Teixeira *et al*. 2022). Li *et al*. (2019) proposes a computer vision method to prevent ship-bridge collisions by using a single shot multibox detector (SSD) Liu *et al*. (2016) and transfer learning to detect ships and calculate their geometric parameters in real-time with monocular vision. Zhang *et al*. (2021) proposed CHPDet, a novel approach for arbitrary-oriented ship detection in remote sensing images. CHPDet formulates ships as rotated boxes with head points, improving angle prediction accuracy and reducing computational cost. Chen *et al*. (2020) designed an augmented ship tracking framework using the kernelized correlation filter (KCF) and curve fitting algorithm to address challenges posed by obstacles in maritime videos. This approach enhanced ship tracking in maritime surveillance, aiding in traffic flow analysis and safety enhancement. Dong *et al*. (2019) proposed a multi-level ship detection algorithm to address challenges in detecting ships from UAV and satellite imagery. The algorithm generated accurate candidate regions using graph-based segmentation and a rotation-invariant descriptor, achieving effectiveness and robustness in detecting ships in optical remote sensing images. Jie *et al*. (2021) proposed an approach for ship detection and tracking in inland waterways to enhance collision avoidance. It improved the YOLOv3 (Farhadi and Redmon 2018) detection algorithm and integrated it with the Deep SORT tracking algorithm. The enhancements to YOLOv3 included optimizing anchor frame initialization, modifying the output classifier, and introducing Soft Non-Maximum Suppression (Bodla *et al*. 2017). Luo *et al*. (2024) designed a novel vessel trajectory identification framework to prevent vessel-bridge collisions. The Co-tracker (Karaev *et al*. 2023) method accurately tracks vessel trajectories by statistically calculating feature point clusters, mitigating the impact of inaccurate target detection. Long Short-Term Memory (LSTM) (Schmidhuber and Hochreiter 1997) and Graph Attention Neural Network (GAT) (Veličković *et al*. 2017) facilitate ship trajectory predictions for anomaly vessel trajectory warnings. However, existing ship trajectory identification methods often require a significant quantity of observations. For example, the approach of Luo *et al*. employed 14 cameras along their 1km testing route to construct a full panoramic view of the target area. In contrast, our 1.5 km testing route is equipped with only 4 cameras, resulting in limited coverage and making the projection process challenging.

In this study, the proposed approach involves Mask R-CNN model and DeepSort algorism to recognize and track the ship trajectory in surveillance video. Utilize the P3P method and Zhang's algorithm to determine the intrinsic matrix and extrinsic matrix for coordinate projection purpose. In addition, we propose an innovative projection process to address the low camera density issue that we calibrate those ship positions out of coverage incorporating the triangulation results within coverage to improve the ship trajectory continuity. This technique will be applied in regions with low camera density or in cases of insufficient observations due to the occurrence of unexpected broken cameras.

1.3 Contribution and scope

In this work, our main contributions are summarized as follows.

- We propose the identification of ship trajectory incorporating Mask R-CNN, DeepSort, and our designed coordinate projection process.
- The novel calibration framework is proposed to address the problem where certain positions along the river are out of coverage by the cameras or the cameras are malfunctioned. We leverage the identification results within the coverage area to

register the ship trajectory with limited or occluded views, thereby reducing the high demand on camera density and mitigating the occurrence of unexpected broken cameras.

Section 2 provides a comprehensive overview of the dataset, including chessboard, point pairs between camera and real world, the ship image dataset, and the test video. Section 3 provides our ship trajectory identification framework, including our training process of the Mask R-CNN model and how we design the calibration process. Section 4 presents the results, offering the ship trajectory we predict in XY-axis, and compare the difference before and after our proposed calibration process.

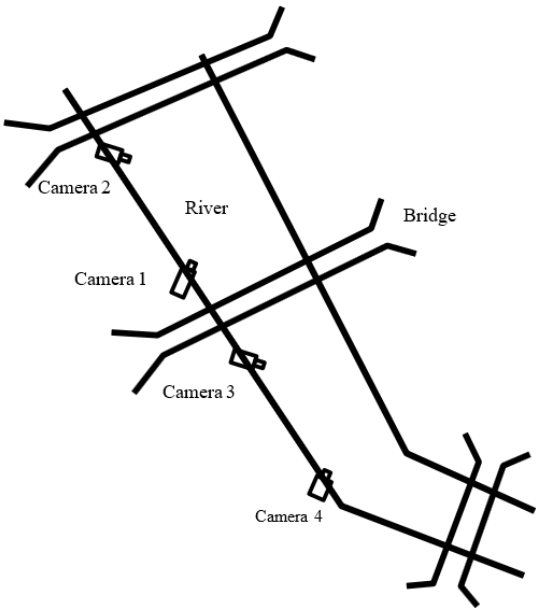


Fig. 1 Relative position of the cameras, there are two pairs of cameras that monitor overlapping regions. However, there are areas that lack concurrent monitoring by two cameras, and in certain cases, have no camera coverage at all

The concluding Section 5 summarizes our contributions, findings, and limitations of this study.

2. Dataset

In this study, we aim to determine ship trajectories by using four surveillance videos captured simultaneously by using four cameras installed at different positions. The relative positions of the cameras are depicted in Fig. 1. There are two pairs of cameras that monitor overlapping regions. This enables us to determine the ship position using triangulation based on the information from two cameras. However, challenges arise in areas not concurrently monitored by two cameras, and in some cases, even have no camera coverage at all. Particularly problematic is the region beneath the bridge, where obstacles such as bridge pillars can significantly impede monitoring. Addressing this issue poses a substantial challenge that requires careful consideration and innovative solutions.

The dataset used herein was provided during the Third International Competition of Structural Health Monitoring (IC-SHM 2022). It consists of four parts:

1. Chessboard images captured using four cameras are shown in Fig. 2. In total, there are four sets of images, and each set contains 20 images. The camera intrinsic matrix can be determined using these images.
2. The ship images are presented in Fig. 3. The dataset contains 1102 images with the ship in the frame; these images were captured from the river bank. The size of these images is 1920 × 1080. However, many of these images are similar or are obscured because of trees along the river bank. We selected and labeled 443 of these images as the training set to train our object detector.
3. Several pixel coordinates of each camera and their corresponding real-world coordinates comprise the third part, and they are used to find the extrinsic matrix of the cameras.

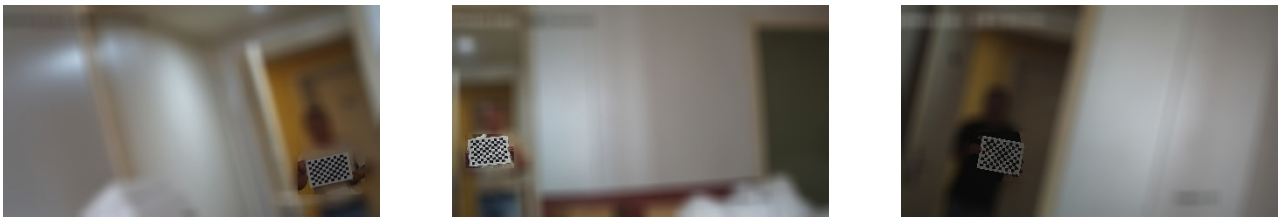


Fig. 2 Chessboard images. The camera intrinsic matrix can be determined using these images.

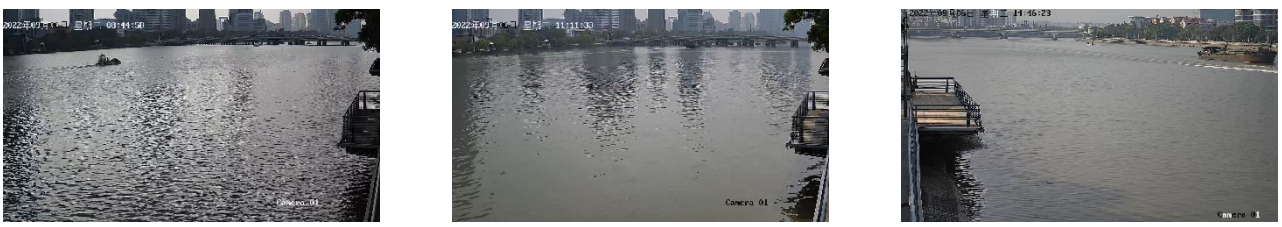


Fig. 3 Ship images

4. Seven groups of test videos recorded simultaneously by using the four cameras are used to test the effectiveness of the ship track recognition program. The size of these videos is 2560 × 1440.

3. Methodology

As discussed in Section 2, during the ship trajectory, certain ship positions may only be captured by a single camera, posing a challenge for conventional triangulation methods in determining the ship coordinates at those instances. To address this issue, we designed an algorithm aimed at approximating the potential position of the target.

Then, we proposed a novel calibration approach to refine the coordinates, ensuring consistency with results obtained through traditional triangulation methods, thus minimizing discrepancies in the calculated outcomes. The overall trajectory-generation process illustrated in Fig. 4 initiates with the acquisition of the ship position. This is achieved through an object detector and object tracker trained on ship images. Subsequently, the intrinsic matrix is determined utilizing Zhang's algorithm (Zhang 2000). To establish the extrinsic matrix of each camera, the perspective-three-point (P3P) method (Gao *et al*. 2003) is employed. Both these metrics utilize chessboard images and three-dimensional (3D)-two-dimensional (2D) point pairs. Finally, we leverage this information and calculate the target ship trajectory using our proposed coordinate projection method. The subsequent section will provide detailed introductions to each component designed in this project.

3.1 Coordinate projection

The four monitors will be split into two groups. Two monitors in each group overlap to monitor the same area as Fig. 1 shown. Therefore, in most cases the ship will be captured by two monitors at the same time, and two observations will be obtained. Then we can leverage these observations to implement coordinate projection using triangulation. This process entails calculating the intersection of lines or rays originating from the camera centres to the corresponding image points. Through this, it can estimate the ship spatial location in a 3D coordinate system. The corresponding equations are represented by Eqs. (1) and (2) as follows

$$X'_{ci} = \frac{R^i_{11}X_w + R^i_{12}Y_w + R^i_{13}Z_w + T^i_x}{R^i_{31}X_w + R^i_{32}Y_w + R^i_{33}Z_w + T^i_z} \tag{1}$$

$$Y'_{ci} = \frac{R^i_{21}X_w + R^i_{22}Y_w + R^i_{23}Z_w + T^i_x}{R^i_{31}X_w + R^i_{32}Y_w + R^i_{33}Z_w + T^i_z} \tag{2}$$

where i = 1, 2, totally 4 equations determine 3 unknown values using singular value decomposition (SVD) to solve Ax = 0 to obtain the world coordinates (X_W, Y_W, Z_W).

However, challenges arise when only one monitor can capture the position of a ship at certain position in time due to factors such as poor weather, obstruction by bridge pillars, and the broken of monitors. In such scenarios, traditional triangulation methods, which require a minimum of two observations, become impractical. To address the issue, we design a novel projection process aimed at refining approximate predictions to closely align with the

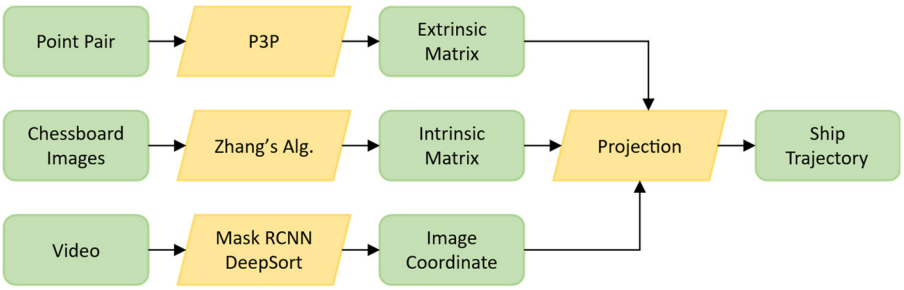


Fig. 4 The trajectory-generation process. We obtain the ship position by using an object detector and object tracker. Then, we determine the intrinsic matrix and extrinsic matrix of each camera by using the chessboard images and the three-dimensional (3D)-two-dimensional (2D) point pairs. Thereafter, we can determine the real-world path of the ship

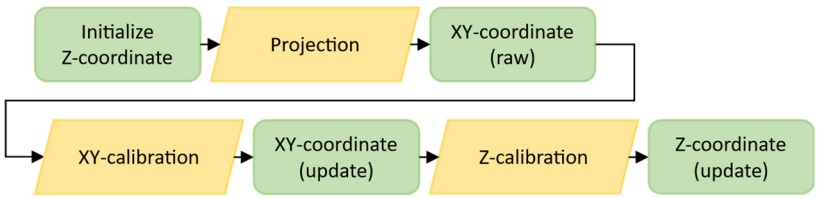


Fig. 5 The calibration process involves initializing the Z-coordinate with the latest reported value from the results of triangulation. Then, we project the XY-coordinate under the assumption Z-coordinate, and calibrate it, and use the updated X-coordinate to calibrate the Z-coordinate for calibrate the next ship position with only one observation. This sequential process ensures accurate calibration of both XY and Z coordinates

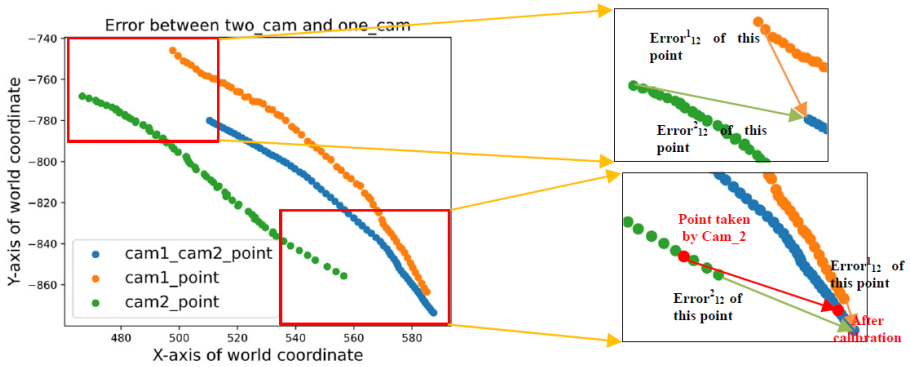


Fig. 6 Example of estimation errors in world X and Y between ship positions with one observation and two observations. Blue points represent the ship positions with two observations (camera 1 and camera 2), orange points represent the same ship trajectory with only one observation from camera 1, and green points represent the positions with one observation from camera 2. $Error^{1}_{12}$ represent the error between orange point and blue point (camera 1 error). $Error^{2}_{12}$ is the same for camera 2. Thus, we can obtain the X and Y errors in each camera using XY-coordinate in corresponding camera

ship positions determined by two observations, shown as Fig. 5. The calibration process involves several steps. Initially, we take one of test video as training data and operate under the assumption that all predicted position triangulated by two observations are accurate. If the current position is captured by a single monitor, the Z-coordinate is initialized using its value from the previous time step, as applied in Eqs. (1) and (2).

Subsequently, a preliminary ship trajectory is derived based on the hypothetical Z-coordinate. Next, the X and Y coordinates are refined through XY-calibration, followed by Z-calibration to update the current Z-coordinate for determining the next position. Finally, a smoothing process is applied to refine the trajectory. The elements of the calibration process are detailed in the subsequent sections.

XY-calibration is just like Fig. 6. The blue points represent the ship positions triangulated by two observations (using camera 1 and 2), the orange points represent the same ship trajectory evaluated using only one observation of camera 1. We utilize the corresponding Z-coordinate of blue points as the hypothetical Z-coordinate, and the green points are similar to orange points but using camera 2 observation. The error between orange points and blue points ($Error^{1}_{12}$), indicating the error of camera 1 because we assume the triangulated positions with two observation is correct. And $Error^{2}_{12}$ are the one for camera 2. Then we define the X and Y errors of point p taken by camera n based on the error value of the nearest sample among all samples captured by camera n to the left of p in the X-coordinate. This process can be similarly applied to cam2-cam4 points and cam3-cam4 points.

Z-calibration, depicted in Fig. 7, aimed at using the ship positions triangulated two observations to calibrate the hypothetical Z-coordinates close to the ground-truth we defined. The blue points represent the ship positions triangulated by two observations (using camera 1 and 2), and the red line represents the curve fitting from those points. Significantly, the ship positions captured by cameras 2 and 4 are situated at a considerable distance from the cameras and in proximity to the bridge pillars. This spatial

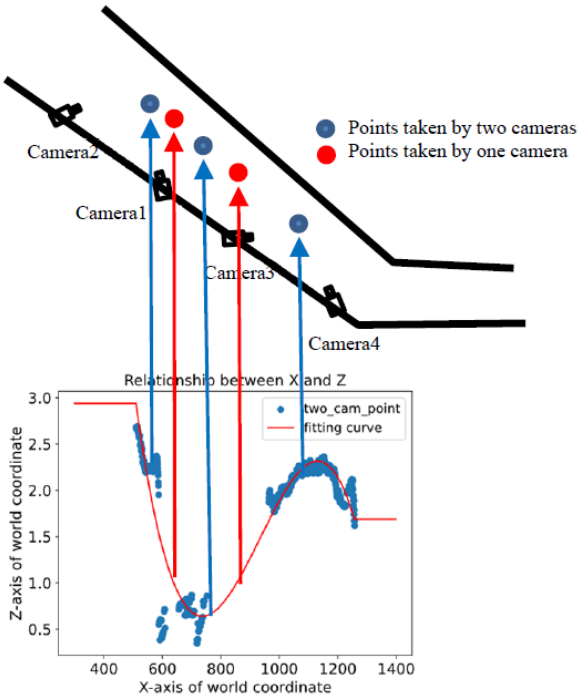


Fig. 7 The relationship between X-coordinate and Z-coordinate, where the blue points represent the ship positions with two observations (camera 1 and camera 2), while the red line represents the curve fitting from those points. Notably, points taken by cameras 2 and 4 are located far from the cameras and near the bridge pillars, introducing some error that causes a sudden drop in the Z-coordinate compared to the ship trajectory generated by the other two pairs of cameras. By using this curve, we can estimate the Z-coordinates of the ship position with one observation based on their X-coordinates

arrangement introduces an error leading to a sudden drop in the Z-coordinate compared to the ship trajectory generated by the other two pairs of cameras. We can estimate the error

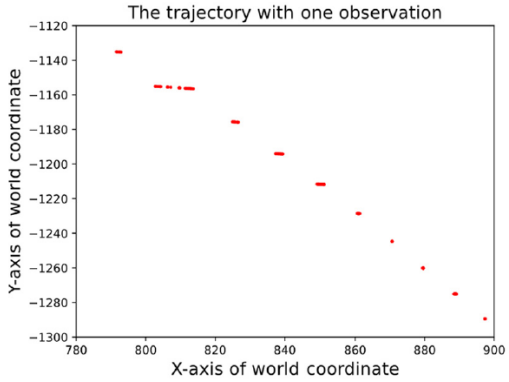


Fig. 8 The ship trajectory with one observation. We design a Gaussian filter for smoothing it

of the hypothetical Z-coordinates using the updated X-coordinates by the curve and calibrate it. Then we can refine the current Z-coordinate and obtain next point based on it.

Ship trajectory smoothing designed to tackle the discontinuity issue resulted from the absence of observed depth information, as illustrated in Fig. 8. Thus, we employ Gaussian smoothing across the entire ship trajectory. The Gaussian filter, represented by Eq. (3), is designed as follow

$$g(x) = \frac{1}{\sqrt{2\pi}\sigma} e^{-x^2/(2\sigma^2)} \tag{3}$$

with $\sigma = 3$ to smooth the ship trajectory. It can tune the ship position using the nearby points to smooth the trajectory.

3.2 Instance segmentation

The target position should be located at a certain point in the frame. To this end, we train a Mask R-CNN model as the detector to achieve instance segmentation and use the centroid of each instance as the image coordinates. Then, we split the dataset into 308 training sets and 35 validation sets to avoid model overfitting. To ensure that the proposed model adapts to the effects of rain or light, we use a few data augmentation techniques such as random brightness and random blur. Model training is completed on a Linux server equipped with four Intel Xeon E5-2620 CPUs, 256 GB DDR4 RAM, and eight NVIDIA RTX Quadro 8000 GPUs with 48 GB memory. In terms of software, MMDetection (Chen *et al.* 2019) is used as the implementation framework. Table 1 summarizes the training parameters. In this project, we fine tune the pretrained model provided by MMDetection. The mAP curve on the validation dataset is depicted in Fig. 9, where the blue line and orange line denote the bounding box mAP and segmentation mAP, respectively. The performance of both loss and map improves after fine-tuning. We determine the final checkpoint based on the epoch with the highest segmentation mAP scores on the validation dataset because we only corporate segmentation maps to evaluate ship centroid.

Table 1 Training parameter of the Mask R-CNN model

Training parameter	Value
Optimizer	SGD
Base Learning Rate (LR_{base})	0.0025
LR decay function	$LR = LR_{base} \times \left(1 - \frac{\text{Epoch}_{\text{current}}}{\text{Epoch}_{\max}}\right)^{0.9}$
Maxepoch ($\text{Epoch}_{\max}$)	36
Batch size	2
Backbone	ResNet50-FPN
Framework	Caffe

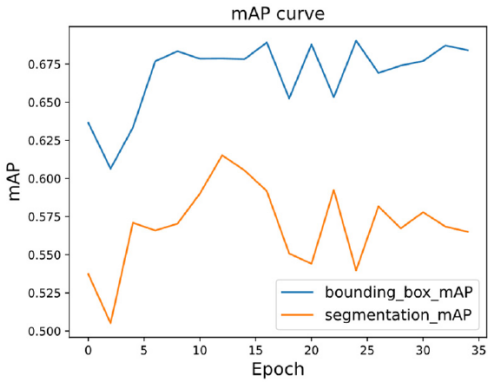


Fig. 9 The mAP curve of the model on validation dataset, where the blue line and orange line denote the bounding box mAP and segmentation mAP, respectively. We determine the final checkpoint based on the epoch with the highest segmentation mAP scores on the validation dataset

3.3 Object tracker

We use DeepSort as the object tracker to match all the objects detected by the object detector in each frame. DeepSort considers motion information by using Eq. (4) as follow

$$d^{(1)}(i,j) = (d_j - y_i)^T S_i^{-1} (d_j - y_i) \tag{4}$$

where $d^{(1)}(i,j)$ denotes the motion distance corresponding to the detection of the i-th track and j-th bounding box. (y_j, S_i) denotes the projection of the i-th track distribution on the measurement space, and d_j denotes the detection of the j-th bounding box. Moreover, DeepSort considers appearance information by using Eq. (5) as follow

$$d^{(2)}(i,j) = \min\{1 - r_j^T r_k^{(i)} | r_k^{(i)} \in R_i\} \tag{5}$$

where $d^{(2)}(i,j)$ denotes the appearance distance corresponding to the detection of the i-th track and j-th bounding box. r_j denotes the descriptor for the detection of each bounding box d_j; DeepSort obtains these descriptors by using a tiny CNN model, and we use mobilenet as the CNN model in this study. Additionally, we maintain a gallery $R_k = \{r_k^{(i)}\}_{k=1}^{L_k}$ of the last $L_k = 100$ associated

appearance descriptors for each track k.

3.4 Intrinsic matrix

We can obtain the intrinsic matrix by means of Zhang's algorithm. This method uses the corners on the chessboard images and solves Eq. (6) as follow

$$s\begin{bmatrix}u\\v\\1\end{bmatrix} = K[R_1\ R_2\ R_3\ T]\begin{bmatrix}X_W\\Y_W\\Z_W\\1\end{bmatrix} \tag{6}$$

to find the intrinsic matrix K. In this equation, (u, v) are the coordinates of the corners on the chessboard images. $[R_1,\ R_2,\ R_3,\ T]$ is the extrinsic matrix. $(X_W,\ Y_W,\ Z_W)$ are the coordinates of the corners on the chessboard images. Thus, Eq. (6) can be simplified to Eq. (7) as follow because Z_W is equal to zero.

$$s\begin{bmatrix}u\\v\\1\end{bmatrix} = K[R_1\ R_2\ T]\begin{bmatrix}X_W\\Y_W\\1\end{bmatrix} \tag{7}$$

3.5 Perspective-n-point

The image coordinates of the control points and the corresponding world coordinate are given. Thus, we have on hand a perspective-n-point problem. We use the perspective-three-point (P3P) method to determine the extrinsic matrix. The method is illustrated in Fig. 10. The image coordinates can be converted to homogeneous coordinates to obtain $\overrightarrow{Oa}$, $\overrightarrow{Ob}$, and $\overrightarrow{Oc}$ by using the calculated intrinsic parameters. Then, $\overrightarrow{OA}$, $\overrightarrow{OB}$, and $\overrightarrow{OC}$ can be obtained using the cosine law, and the camera coordinates of A, B, C can be computed. We can determine the extrinsic matrix by using Eq. (8) as follow

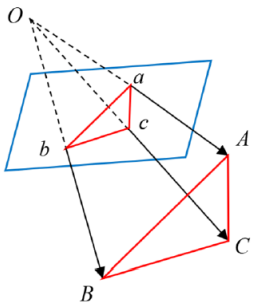


Fig. 10 The P3P method, where the blue parallelogram represents the projection plane

$$\begin{bmatrix}X_W\\Y_W\\Z_W\\1\end{bmatrix} = [R_1\ R_2\ R_3\ T]\begin{bmatrix}X_C\\Y_C\\Z_C\\1\end{bmatrix} \tag{8}$$

where $(X_W,\ Y_W,\ Z_W)$ and $(X_C,\ Y_C,\ Z_C)$ denote world coordinates and camera coordinates, respectively.

3.6 Random sample consensus

The P3P method can determine an extrinsic matrix by using only three control points. However, it is affected by outliers among the control points. We introduce the random sample consensus (RANSAC) (Fischler and Bolles 1981) mechanism to find the best set of control points. The process is illustrated in Fig. 11. We select a set of control points randomly and find an extrinsic matrix. Then, we reproject all 2D coordinates as 3D coordinates by using the extrinsic matrix and select the point for which the distance between the observed and computed point projections is shorter than the threshold distance required to consider a point an inlier. We repeat the process 100 times to find the extrinsic matrix with maximum inliers. In other words, this matrix is the most general for all control points, and its probability of being close to the optimal solution is the highest.

4. Result

4.1 Ship segmentation

The results of our instance segmentation model can be seen in Fig. 12. The model is capable of accurately segmenting both big and small ships, and estimating the centroid of each instance. Additionally, we have addressed the issue of image blurring by implementing data augmentation techniques, which have helped to improve the accuracy of our model.

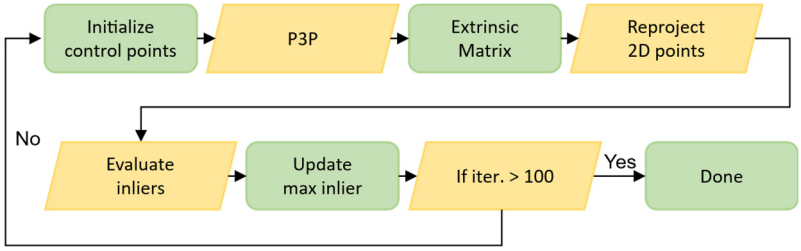
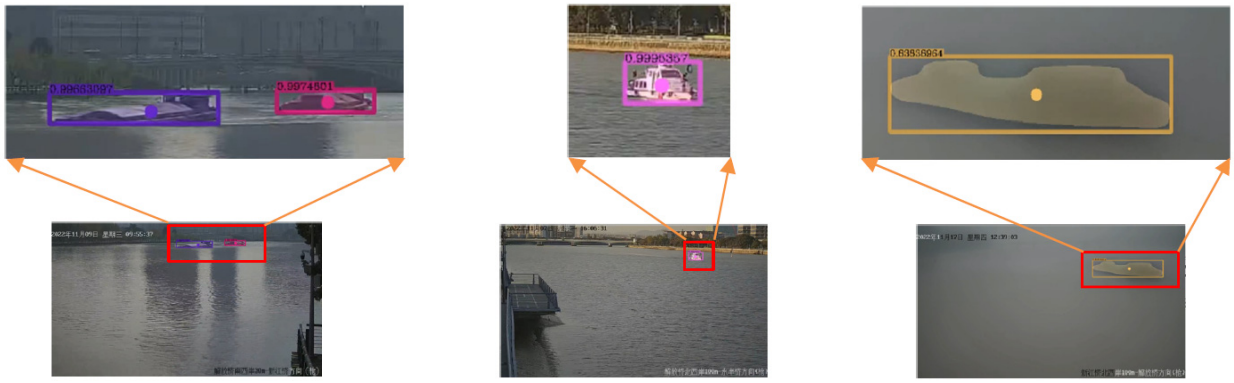
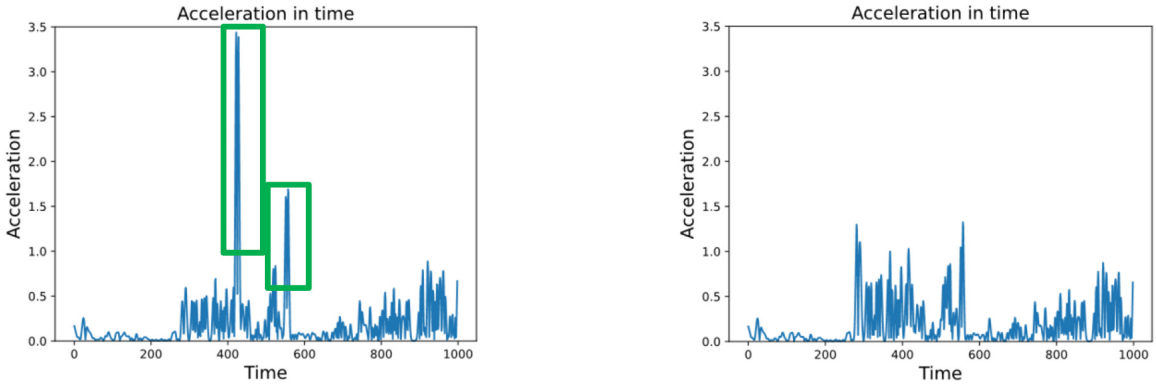


Fig. 11 The RANSAC process. A subset of control points is randomly chosen, and an extrinsic matrix is determined using the P3P method. Subsequently, all 2D coordinates are reprojected as 3D coordinates using the extrinsic matrix. Points are selected based on the criterion that the distance between observed and computed point projections is shorter than a specified threshold, designating them as inliers. This process is iterated 100 times to identify the extrinsic matrix with the maximum number of inliers



(a) (b) (c)

Fig. 12 Result of ship segmentation. Our model is capable of accurately segmenting both (a) big and (b) small ships, even (c) the scene is so blurry



(a) (b)

Fig. 13 The acceleration in time: (a) before Z-calibration; (b) after Z-calibration. The two peaks highlighted in the green rectangle disappeared after calibration

4.2 Coordinate projection

4.2.1 Evaluation metrics

Because of the absence of ground-truth, the metrics inspired from Luo *et al*. (2024) aimed to evaluate two qualities of the prediction ship trajectory:

1. Acceleration variability (AV): The higher acceleration variability indicates the substantial acceleration changes, suggesting an erratic ship trajectory. Thus, we evaluate the acceleration variability using Eq. (9) as follow

$$\frac{Std\ Acceleration}{Mean\ Acceleration} = \frac{\sqrt{\frac{1}{N}\sum_{i=1}^{N}(A_i - \frac{1}{N}\sum_{i=1}^{N}A_i)^2}}{\frac{1}{N}\sum_{i=1}^{N}A_i} \quad (9)$$

A_i represent the acceleration at the i-th observation and N be the total number of observations. Thus, the higher value indicates the higher variability.

2. Directional change (DC): The higher directional change indicates the significant directional shifts, representing a nonsense ship trajectory. Thus, we evaluate the acceleration variability using Eq. (10) as follow

$$DC = \frac{1}{N-1}\sum_{i=1}^{N-1}|\theta_i - \theta_{i-1}| \quad (10)$$

where θ_i is the direction angle at the i-th position.

4.2.2 Testing set

In our coordinate projection process, we require a training set of videos to construct the error map. We specifically select a set of videos capturing a single large ship in favorable weather conditions. This selection is made because such footage typically features fewer obstacles and clearer scenes, enabling the creation of a high-quality error map based on the points captured by two monitors. For testing purposes, we utilize four sets of videos featuring large ships from the seven sets provided by IC-SHM 2022.

4.2.3 Calibration result

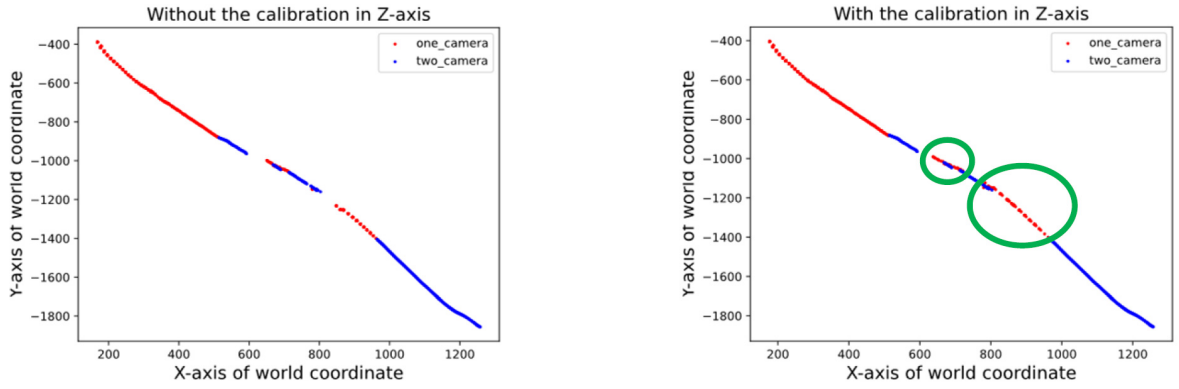
The calibration results are presented in Table 2, where we conducted an ablation study to assess the impact of each element in our proposed method. The first row represents the baseline without any calibration. It is evident that the Gaussian filter significantly reduces the DC score from 0.85 to 0.08 and the AV score from 2.75 to 1.95. Subsequently,

Table 2 Ablation study of calibration results for proposed method. It presents the calibration results for our proposed method, illustrating the effectiveness of each element. The AV score decreases from 2.75 to 1.54 through our calibration method, while the DC score decreases from 0.85 to 0.09. These findings indicate that all elements in this framework contribute to effectively smoothing the ship trajectory

Gaussian	XY-calibration	Z-calibration	Metrics	
			AV	DC (rad)
-	-	-	2.75	0.85
✓	-	-	1.95	0.08
✓	✓	-	1.56	0.07
✓	✓	✓	1.54	0.09

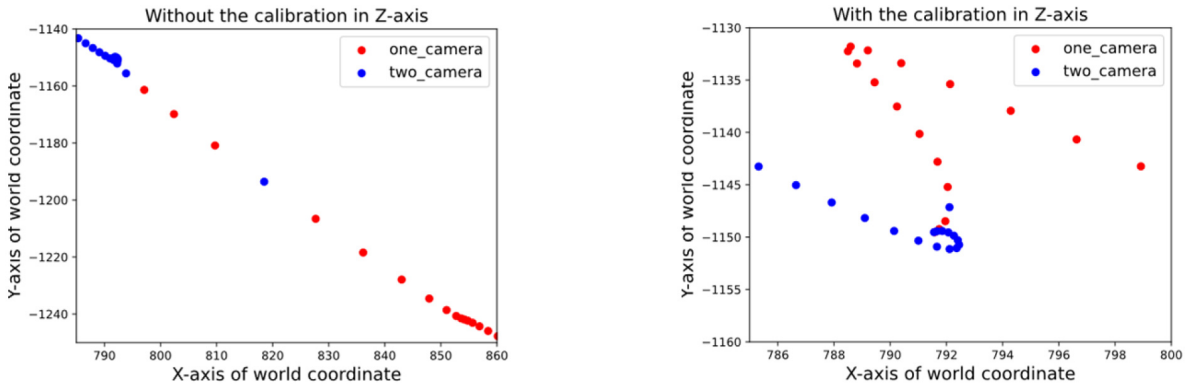
our proposed calibration method in the XY-axis decreases the AV score to 1.56 and the DC score to 0.07. Finally, calibration in the Z-axis achieves the lowest AV score, but a slight increase in the DC score to 0.09.

In order to explain that why the DC score slightly increase after employing calibration in Z-axis, we utilized AV plot and DC plot analysis, as depicted in Fig. 13. Figs. 13(a) and (b) illustrate the acceleration over time before and after implementing calibration in the Z-axis. We observed that the two peaks highlighted in the green rectangle disappeared after calibration, indicating that Z-axis calibration extended the ship positions with one observation. This extension is more evident in Fig. 14. In order to facilitate discussion of the effects of Z-calibration, we differentiated the ship positions captured by one camera or two cameras with different colors. Notably, the ship positions with one observation in green circles at (b) became longer after calibration, resulting in a rise in the AV score. However, this extension also brought some side effects, as shown in Fig. 15. Figs. 15(a) and (b) depict the ship trajectory before and after employing calibration in the Z-axis. We observed that the extension is not always reasonable, as the ship trajectory exhibited a weird track on the X-axis between 600 and 800, near bridges with poor navigation clearance. Thus, the reason of the weird extension is resulted from the Z-coordinate map mentioned in Section 3.1. Calibration based on the wrong map can register a strange track due to spatial arrangement errors. This evidence is depicted in Fig. 16. We can find that except for the unstable ship trajectory in the last one hundred second, the extremely large directional changes all occur



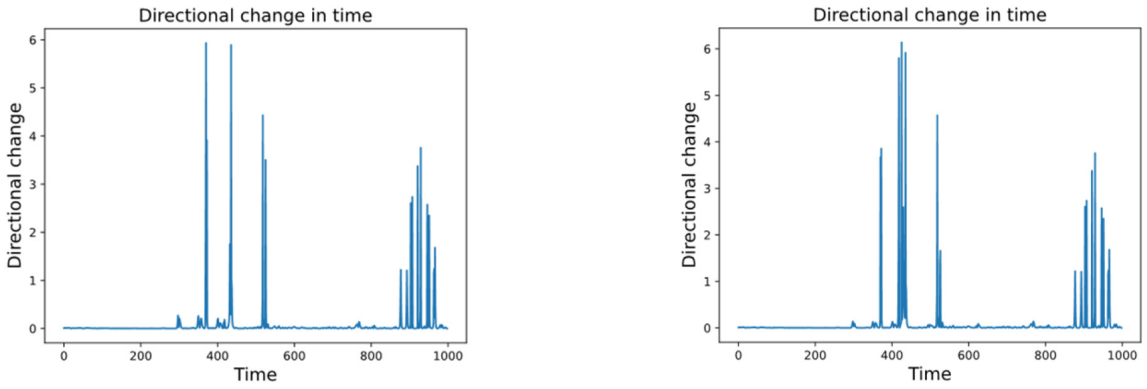
(a) (b)

Fig. 14 The result of calibrating Z-coordinate: (a) before Z-calibration; (b) after Z-calibration. We can find that the ship trajectory with one observation in green circles at (b) became longer after calibration



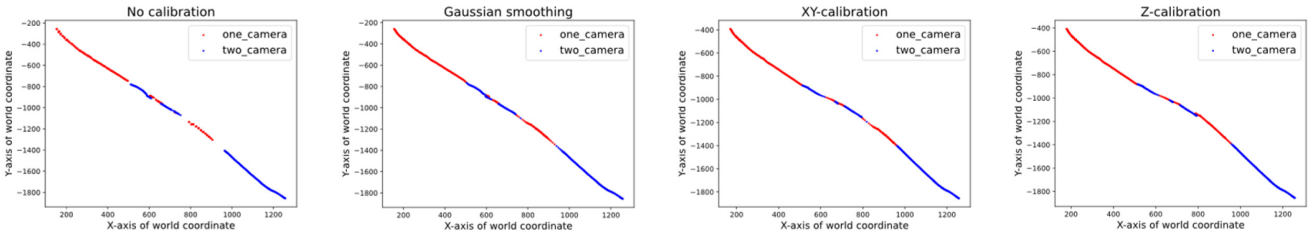
(a) (b)

Fig. 15 The weird ship trajectory after calibrating Z-coordinate: (a) before calibration; (b) after calibration. The ship trajectory shown a weird track on the X-axis between 600 and 800, where is near bridges



(a) (b)

Fig. 16 The directional change in time: (a) before calibration (b) after calibration. Except for the unstable ship trajectory in the last one hundred second, the extremely large directional changes all occur near the bridges



(a) (b) (c) (d)

Fig. 17 The visualization results: (a) to (d) corresponding to the settings from top to bottom in Table 2. The large gaps decrease after implementing our calibration process

(a) (b)

Fig. 18 The limitations of the proposed framework: (a) Low illumination case and (b) small ships case

near the bridges. In other words, if an additional monitor near the bridge captures clear video for generating a better Z-coordinate map, the issue can be mitigated. However, the calibration in the Z-axis can still improve the smoothness of the ship trajectory by reducing the AV score.

The visualization results are presented in Fig. 17, with images from left to right corresponding to the settings from top to bottom in Table 2. It is evident that the large gaps in the ship trajectory decrease after implementing our calibration process. In the final result, there are almost no gaps, but some overlaps near the bridges can be observed.

4.3 Limitation

First, the Mask R-CNN model exhibits suboptimal performance in dark scenes owing to the absence of a dataset specifically tailored for low-illumination conditions. We have attempted to address this issue by implementing various computer vision methods, such as adding synthetic data and applying random brightness data augmentation. However, the model still exhibits inferior performance. The condition could be mitigated providing that the dataset includes some more dark scene images. Second, the calibration model also exhibits inferior performance when detecting small ships. The predicted trajectory of small ship video is depicted in Fig. 18(b). We can find that the ship turned around between X-axis coordinates of 900 and 1000, which did not occur in the actual video. Future studies are necessary to address the aforementioned issues.

Furthermore, with poor navigation clearance near the bridges, the Z-coordinate map shows a significant drop in

that area, indicating an error. Therefore, the Z-calibration makes the ship trajectory extend wrongly in this area, as it is based on the erroneous Z-coordinate map. However, this issue can be mitigated by adjusting the monitor arrangement to improve video quality and reconstruct a more accurate Z-coordinate map.

Consequently, due to the lack of ground-truth, the indicators employed focus on the continuity of the ship trajectory. Future endeavor aims to collect ground-truth to validate the predicted trajectories against actual ship movements, thereby improving the reliability of the model.

5. Conclusions

In this study, a deep learning-based object tracking framework is proposed to reconstruct the ship trajectory from videos. The framework is evaluated in several cases, and the major findings and limitations of our study are as follows:

- We identify the ship trajectory based on deep learning-based segmentation models and computer vision techniques.
- Since the low observation density due to the lack of camera or the unexpected malfunctioned cameras, the ship may not be captured by two cameras. We then propose a novel calibration framework capable of estimating the ship trajectory out of coverage using the ship trajectory in coverage area. Additionally, we design a calibration process to calibrate ship trajectory with one observation using those with two cameras. After implementation, we achieve a boost in testing performance, where the AV and DC scores are reduced from 2.75 to 1.54 and 0.85 to 0.09. respectively.
- We address the image blurring issue by employing data augmentation techniques.
- Future studies will focus on enhancing the identification of small ships and ships in dark scenes. Additionally, a more accurate Z-coordinate mapping is required by adjusting the monitor arrangements to improve the video quality. Furthermore, ground-truth data will be collected to enable more comprehensive evaluations of model performance.

Acknowledgments

The authors would like to appreciate the organization of the IC-SHM 2022: ANCRiSST, University of Illinois at Urbana-Champaign, Harbin Institute of Technology, Zhejiang University, and University of Houston for providing the valuable data used in this study.

References

Avidan, S. and Shashua, A. (2000), "Trajectory triangulation: 3D reconstruction of moving points from a monocular image sequence", *IEEE Transact. Pattern Anal. Mach. Intell.*, **22**(4), 348-357. https://doi.org/10.1109/34.84537

Bewley, A., Ge, Z., Ott, L., Ramos, F. and Upcroft, B. (2016), "Simple online and realtime tracking", In: *2016 IEEE International Conference on Image Processing* (*ICIP*), Phoenix, AZ, USA, September.

Bodla, N., Singh, B., Chellappa, R. and Davis, L.S. (2017), "Soft-NMS--improving object detection with one line of code", *Proceedings of the IEEE International Conference on Computer Vision*, Venice, Italy, October.

Campbell, S., Naeem, W. and Irwin, G.W. (2012), "A review on improving the autonomy of unmanned surface vehicles through intelligent collision avoidance manoeuvres", *Annual Rev. Control*, **36**(2), 267-283. https://doi.org/10.1016/j.arcontrol.2012.09.008

Chauvin, C., Lardjane, S., Morel, G., Clostermann, J.P. and Langard, B. (2013), "Human and organisational factors in maritime accidents: Analysis of collisions at sea using the HFACS", *Accident Anal. Prevent.*, **59**, 26-37. https://doi.org/10.1016/j.aap.2013.05.006

Chen, K., Wang, J., Pang, J., Cao, Y., Xiong, Y., Li, X., Sun, S., Feng, W., Liu, Z. and Xu, J. (2019), "MMDetection: Open mmlab detection toolbox and benchmark", arXiv preprint arXiv:1906.07155. https://doi.org/10.1016/j.simpa.2021.100081

Chen, X., Xu, X., Yang, Y., Wu, H., Tang, J. and Zhao, J. (2020), "Augmented ship tracking under occlusion conditions from maritime surveillance videos", *IEEE Access*, **8**, 42884-42897. https://doi.org/10.1109/ACCESS.2020.2978054

Dong, C., Liu, J., Xu, F. and Liu, C. (2019), "Ship detection from optical remote sensing images using multi-scale analysis and Fourier HOG descriptor", *Remote Sens.*, **11**(13), 1529. https://doi.org/10.3390/rs11131529

Everingham, M., Van Gool, L., Williams, C.K., Winn, J. and Zisserman, A. (2010), "The pascal visual object classes (voc) challenge", *Int. J. Comput. Vis.*, **88**, 303-338. https://doi.org/10.1007/s11263-009-0275-4

Farhadi, A. and Redmon, J. (2018), "Yolov3: An incremental improvement", arXiv preprint arXiv:1804.02767. https://doi.org/10.4324/9780203978948-13

Fischler, M.A. and Bolles, R.C. (1981), "Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography", *Commun. ACM*, **24**(6), 381-395. https://doi.org/10.1016/B978-0-08-051581-6.50070-2

Gao, X.S., Hou, X.R., Tang, J. and Cheng, H.F. (2003), "Complete solution classification for the perspective-three-point problem", *IEEE Transact. Pattern Anal. Mach. Intell.*, **25**(8), 930-943. https://doi.org/10.1109/TPAMI.2003.1217599

Girshick, R. (2015), "Fast r-cnn", *Proceedings of the IEEE International Conference on Computer Vision*, Santiago, Chile, December.

Girshick, R., Donahue, J., Darrell, T. and Malik, J. (2014), "Rich feature hierarchies for accurate object detection and semantic segmentation", *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, Columbus, OH, USA, June.

Gupta, V., Gupta, M. and Singla, P. (2021), "Ship detection from highly cluttered images using convolutional neural network", *Wireless Personal Commun.*, **121**(1), 287-305. https://doi.org/10.1007/s11277-021-08635-5

Hartley, R.I. and Sturm, P. (1997), "Triangulation", *Comput. Vis. Image Underst.*, **68**(2), 146-157. https://doi.org/10.1515/9783110326833.69

He, K., Gkioxari, G., Dollár, P. and Girshick, R. (2017), "Mask r-cnn", *Proceedings of the IEEE International Conference on Computer Vision*, Venice, Italy, October.

Jeong, S. and Kim, T.W. (2023), "Generating a path-search graph based on ship-trajectory data: Route search via dynamic programming for autonomous ships", *Ocean Eng.*, **283**, 114503.

https://doi.org/10.1016/j.oceaneng.2020.108242
Jie, Y., Leonidas, L., Mumtaz, F. and Ali, M. (2021), "Ship detection and tracking in inland waterways using improved YOLOv3 and Deep SORT", *Symmetry*, **13**(2), 308. https://doi.org/10.3390/sym13020308
Karaev, N., Rocco, I., Graham, B., Neverova, N., Vedaldi, A. and Rupprecht, C. (2023), "Cotracker: It is better to track together", arXiv preprint arXiv:2307.07635. https://doi.org/10.2307/j.ctv11cvxt5
Kartal, M. and Duman, O. (2019), "Ship detection from optical satellite images with deep learning", In: *2019 9th International Conference on Recent Advances in Space Technologies* (*RAST*), Istanbul, Turkey, June.
Krizhevsky, A., Sutskever, I. and Hinton, G.E. (2012), "Imagenet classification with deep convolutional neural networks", *Adv. Neural Inform. Process. Syst.*, **25**. https://doi.org/10.1145/3065386
Li, S., Guo, Y., Xu, Y. and Li, Z. (2019), "Real-time geometry identification of moving ships by computer vision techniques in bridge area", *Smart Struct. Syst., Int. J.*, **23**(4), 359-371. https://doi.org/10.12989/sss.2019.23.4.359
Liu, W., Anguelov, D., Erhan, D., Szegedy, C., Reed, S., Fu, C.Y. and Berg, A.C. (2016), "Ssd: Single shot multibox detector", In: *Computer Vision–ECCV 2016*: *14th European Conference*, Amsterdam, The Netherlands, October.
Luo, W., Xia, Y. and He, T. (2024), "Video-Based Identification and Prediction Techniques for Stable Vessel Trajectories in Bridge Areas", *Sensors*, **24**(2), 372. https://doi.org/10.1111/j.1747-1567.2001.tb00047.x
Polvara, R., Sharma, S., Wan, J., Manning, A. and Sutton, R. (2018), "Obstacle avoidance approaches for autonomous navigation of unmanned surface vehicles", *J. Navigat.*, **71**(1), 241-256. https://doi.org/10.1017/S0373463317000753
Ren, S., He, K., Girshick, R. and Sun, J. (2015), "Faster r-cnn: Towards real-time object detection with region proposal networks", *Adv. Neural Inform. Process. Syst.*, **28**. https://doi.org/10.1109/TPAMI.2016.2577031
Schmidhuber, J. and Hochreiter, S. (1997), "Long short-term memory", *Neural Computat.*, **9**(8), 1735-1780. https://doi.org/10.1142/97898127765630022
Štepec, D., Martinčič, T. and Skočaj, D. (2019), "Automated system for ship detection from medium resolution satellite optical imagery", In: *OCEANS 2019 MTS/IEEE SEATTLE*, Seattle, WA, USA, October.
Stofa, M.M., Zulkifley, M.A. and Zaki, S.Z.M. (2020), "A deep learning approach to ship detection using satellite imagery", In: *IOP Conference Series*: *Earth and Environmental Science*, Kuala Lumpur, Malaysia, July.
Teixeira, E., Araujo, B., Costa, V., Mafra, S. and Figueiredo, F. (2022), "Literature review on ship localization, classification, and detection methods based on optical sensors and neural networks", *Sensors*, **22**(18), 6879. https://doi.org/10.3390/s22186879
Veličković, P., Cucurull, G., Casanova, A., Romero, A., Lio, P. and Bengio, Y. (2017), "Graph attention networks", arXiv preprint arXiv:1710.10903. https://doi.org/10.1016/j.neunet.2022.11.021
Wang, J., Song, Y., Wang, W. and Chen, C. (2019), "Evaluation of flexible floating anti-collision device subjected to ship impact using finite-element method", *Ocean Eng.*, **178**, 321-330. https://doi.org/10.1016/j.oceaneng.2019.03.005
Wang, F., Chang, H.J., Ma, B.H., Wang, Y.G., Yang, L.M., Liu, J. and Dong, X.L. (2023), "Flexible guided anti-collision device for bridge pier protection against ship collision: Numerical simulation and ship collision field test", *Ocean Eng.*, **271**, 113696. https://doi.org/10.1002/stco.200910004
Wojke, N., Bewley, A. and Paulus, D. (2017), "Simple online and realtime tracking with a deep association metric", In: *2017 IEEE International Conference on Image Processing* (*ICIP*), Beijing, China, September.
Yildirim, E. and Kavzoglu, T. (2021), "Ship detection in optical remote sensing images using YOLOv4 and Tiny YOLOv4", *Proceedings of the International Conference on Smart City Applications*, Safranbolu, Turkey, November.
Zhang, Z. (2000), "A flexible new technique for camera calibration", *IEEE Transact. Pattern Anal. Mach. Intell.*, **22**(11), 1330-1334. https://doi.org/10.1007/s00607-019-00723-6
Zhang, F., Wang, X., Zhou, S., Wang, Y. and Hou, Y. (2021), "Arbitrary-oriented ship detection through center-head point extraction", *IEEE Transact. Geosci. Remote Sens.*, **60**, 1-14. https://doi.org/10.1109/TGRS.2021.312041
Zhao, C., Cao, X. and Ren, Y. (2023), "Risk analysis of bridge ship collision based on AIS data model and nonlinear finite element", *Nonlinear Eng.*, **12**(1), 20220324. https://doi.org/10.1016/j.engstruct.2012.03.026
Zwemer, M.H., Wijnhoven, R.G. and Peter HN de With (2018), "Ship Detection in Harbour Surveillance based on Large-Scale Data and CNNs", In: *VISIGRAPP* (5: VISAPP), pp. 153-160. https://doi.org/10.1117/12.2000452