공학석사 학위논문

A Method for Detection and Tracking of Maritime Obstacles Based on Multi-Video

다중 영상 기반 해상 장애물 탐지 및 추적 방법

2023년 2월

서울대학교 대학원
조선해양공학과
박 정 호

A Method for Detection and Tracking of Maritime Obstacles Based on Multi-Video

지도 교수 노 명 일

이 논문을 공학석사 학위논문으로 제출함
2022년 10월

서울대학교 대학원
조선해양공학과
박 정 호

박정호의 공학석사 학위논문을 인준함
2023년 2월

위 원 장__________우 종 훈__________
부위원장__________노 명 일__________
위 원__________남 보 우__________

Table of Contents

Figures

Tables

Equations

Abstract

A method for detection and tracking of maritime obstacles based on multi-video

Jeong-Ho Park
Naval Architecture and Ocean Engineering
The Graduate School
Seoul National University

Among the causes of marine accidents, human error accounts for a relatively high rate, and accordingly, the need for an autonomous recognition technology for recognizing the surroundings is emerging. Research on autonomous recognition technology using traditional recognition sensors such as Automatic Identification Systems (AIS) and Radio Detection and Ranging (RADAR) is being actively conducted, but there are clear limits. Therefore, we tried to develop a new cognitive technology to replace them.

In this paper, we proposed an autonomous recognition technology using a camera to supplement the limitations of traditional cognitive sensors and replace human vision. First, the YOLOv5 algorithm, a real-time object detection algorithm based on camera images, was improved to increase obstacle detection accuracy. Then, a position transformation algorithm estimated the relative position of the detected obstacle. Based on the relative position of the obstacle, we proposed an adaptive extended Kalman filter to estimate the motion of the obstacle, such as trajectory, Course Over Ground (COG), and Speed Over Ground (SOG). In addition, assuming that USV is operated for strategic purposes and mutual communication and cooperation are possible, sensor fusion between data tracked by different cameras was performed to increase the accuracy of tracking data. It was confirmed that more accurate tracking data could be obtained by fusing the tracked data from several cameras to improve tracking accuracy or compensate for the disadvantages that occur in the tracking process of individual cameras.

Keywords: Object detection (장애물 탐지), Object tracking (장애물 추적), Adaptive tracking filter (적응형 추적 필터), Data association (데이터 연관), Sensor fusion (센서 퓨전)

Student number: 2021-21642

1. Introduction

1.1. Research background

Research on autonomous ship navigation systems is being actively conducted to solve maritime accidents caused by human negligence and the problem of a gradually decreasing crew. The autonomous navigation system of a ship is composed of various technologies such as obstacle recognition, obstacle avoidance, and path following. Among them, accurate obstacle recognition technology must be preceded.

In coastal areas where Unmanned Surface Vehicles (USVs) are operated, many maritime obstacles are fatal to USVs, such as small boats without an Automatic Identification System (AIS). Marine obstacles in coastal areas are often smaller than those in the open ocean. They have many variables, making them difficult to detect with traditional recognition systems such as AIS and Radio Detection And Ranging (RADAR). RADAR mounted on USVs can detect obstacles in a wide range, but there is a blind spot within 150m around the boat due to ocean reflection. For large vessels, blind spots do not have a significant effect, but for relatively small USVs, RADAR blind spots are fatal. In addition, Light Detection And Ranging (LiDAR), which is currently being actively researched as one of the new cognitive technologies, is used as a means for short-range detection as opposed to RADAR. However, the price of the equipment increases rapidly as the detection distance increases. Therefore, even if the corresponding recognition technology is mounted on a boat, it still has to rely on human vision to recognize such obstacles.

In this paper, to compensate for the shortcomings of traditional recognition technologies, we proposed a recognition technology that detects and tracks obstacles around the boat within the human visual range using a camera. As shown in Figure 1, the cognitive technology using a camera has the advantage of detecting short-range obstacles that are difficult to detect with AIS and RADAR and utilizing various visual features that can be obtained from images. In addition, since each type of camera has different obstacle detection and tracking characteristics, it is possible to fuse data tracked by Electro-Optical (EO) and Infrared (IR) cameras, respectively, or data tracked from different viewpoints. It was fused to increase the tracking accuracy.

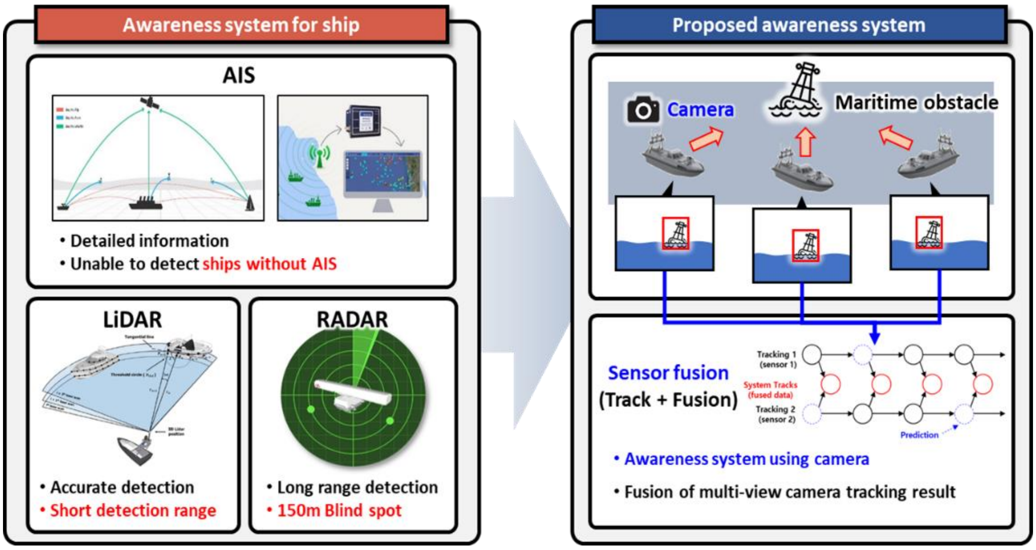


Figure 1. Proposed awareness system using cameras

1.2. Related works

Research to develop cognitive technology using cameras or to utilize various visual information obtained from camera images in various fields is being conducted regardless of the field. In particular, many studies have been conducted to improve detection accuracy by introducing the attention algorithm to the traditional convolutional neural network (CNN), a representative image analysis algorithm. Zhu et al. [1] improved the detection accuracy of the You Only Look Once v5 (YOLOv5, GitHub [2]) algorithm while maintaining the inference speed by using Convolutional Block Attention Module (CBAM, Woo et al. [3]) and Efficient Channel Attention Network (ECA-Net, Wang et al. [4]). The study detected target rock in planetary images, and the detection accuracy was improved by about 3.4% compared to before improvement.

Fu et al. [5] proposed SSIM-Weighted Multiple Instance Learning (SSIM-WMIL) for tracking a specific object in an adjacent frame and verified it on BlueCar4 video data, a road-driving image. First, the SSIM-based classifier is trained by selecting positive and negative samples for objects in the frame of the past time. Next, they selected the candidate most similar to the object among the candidates of the object extracted from the current frame, and tracking was performed using the trained classifier.

As one of the methods of utilizing visual information obtained from images, research on an algorithm that can maintain similar detection or labeling performance even if the domains of the data set are different (domain shifted) is being conducted. Rezaeianaran et al. [6] proposed a Visually Similar Group Alignment (ViSGA) algorithm that can adapt to changes between different domains through visual similarity-based clustering and adversarial training. They solved the problem that detection performance varies depending on the domain of the data set.

As interest in autonomous navigation increases, research on recognizing and tracking obstacles around boats using cameras is being conducted in the field of shipbuilding. Zhang et al. [7] proposed the improved-YOLOv3 as a maritime obstacle detection algorithm that improved the network structure of the YOLOv3 (Redmon et al. [8]) algorithm. They improved detection accuracy by 0.79% based on mean Average Precision (mAP).

Han et al. [9] detected a maritime obstacle using the Single Shot multi-box Detector (SSD) algorithm. They estimated the motion data of the obstacle by tracking it based on the Extended Kalman Filter (EKF). In addition, they attempted to improve the tracking accuracy by fusing camera image-based tracking data with radar-based tracking data acquired similarly. Lee et al. [10] conducted a similar study. They used the YOLOv3 algorithm to detect maritime obstacles from EO, IR, and panorama (wide EO) camera images acquired by mounting them on a small boat. For training the obstacle detection algorithm, they create virtual ocean images. Then, using the detected bounding box, they estimated the motion of the obstacles by tracking through an EKF-based tracking algorithm.

In this paper, we tried to analyze the characteristics of the camera, the most important sensor in the proposed obstacle recognition technology, and the obstacle detection algorithm using the camera and develop the most suitable detection and tracking algorithm. First, we implemented an algorithm that enables fast and accurate detection even on hardware with poor performance using the YOLOv5 algorithm and the CBAM algorithm for detection.

Second, in the data association of detection results and tracking data, existing methods using visual features or distances between bounding boxes do not consider obstacles' motion. Therefore, accurate matching is difficult when many obstacles are densely clustered or intersected. Therefore, it was confirmed that there are limitations. So, in this paper, based on the estimated motion data of the obstacle, the position of the next time was predicted to enable a more accurate association and used in the data association process.

In obstacle tracking, we proposed an Adaptive Extended Kalman Filter (AEKF) designed to appropriately change the error covariance of sensor measurements among EKF parameters according to the detection results. It reflects the detection uncertainty of the obstacle detection algorithm. In addition, a more robust and accurate tracking algorithm was developed compared to operating a single camera by fusing data tracked by multiple types of cameras or cameras mounted on multiple boats through sensor fusion. A summary of related research and this paper is shown in Table 1.

Table 1. Summary of related works and this study

Related works	Image	Detection algorithm	Data association	Obstacle tracking	Sensor fusion
Rezaeianaran et al. (2021)	Single car image	CNN (Faster-RCNN)	Visual feature (ViSGA)	X	X
Fu et al. (2019)	Single car image	-	Visual feature (SSIM)	O (on image)	X
Zhu et al. (2021)	Single planetary image	CNN (YOLOv5) + CBAM/ECA-Net	X	X	X
Zhang et al. (2020)	Single USV image	CNN (improved YOLOv3)	-	O	X
Han et al. (2020)	Multi USV images (EO, IR)	CNN (SSD)	Bounding box distance	O (EKF)	O (camera-RADAR)
Lee et al. (2021)	Multi USV images (EO, IR, panorama)	CNN (YOLOv3)	Bounding box distance	O (EKF)	X
This study (2023)	Multi USV images (EO, IR, panorama)	CNN (YOLOv5) + CBAM	Bounding box estimation + bounding box distance	O (AEKF)	O (multiple cameras)

1.3. Process of the proposed recognition system

This paper proposes an obstacle recognition algorithm using a camera, as shown in Figure 2. First, an obstacle detection algorithm was constructed by adding a CBAM module to the YOLOv5 algorithm. Then, the proposed obstacle detection algorithm was trained by classifying several maritime obstacle-related images acquired in the actual sea into training and validation images.

When an image comes in from the camera, the obstacle detection algorithm detects obstacles every time and extracts a bounding box. At this time, since the extracted bounding box of the obstacle represents only position information on the image plane, position transformation was performed to estimate the motion of the obstacle.

The obstacle tracking algorithm estimates the motion of the obstacle by performing tracking based on the calculated relative position of the obstacle. For tracking, the adaptive extended Kalman filter proposed in this paper was used, and the motion data of the obstacle, such as trajectory, COG, and SOG, was estimated. Furthermore, the information tracked by multiple cameras for the same obstacle was fused using a sensor fusion algorithm to improve tracking accuracy. Finally, the effectiveness of the algorithm proposed in this paper was verified based on the images and navigation data acquired from the actual sea.

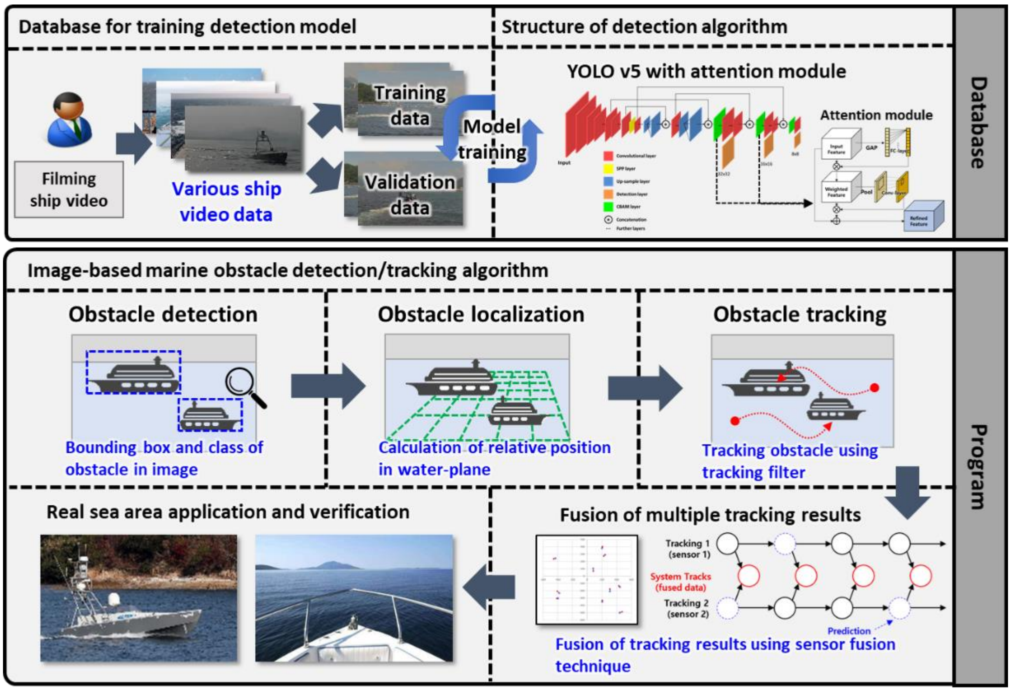

Database for training detection model
Filming ship video
Various ship video data
Training data
Validation data
Model training
Structure of detection algorithm
YOLO v5 with attention module
Input
Convolutional layer
SPP layer
Up-sample layer
Detection layer
CBAM layer
Concatenation
Further layers
32x32
16x16
8x8
Attention module
Input Feature
GAP
FC layer
Weighted Feature
Pool
Conv layer
Refined Feature
Database
Image-based marine obstacle detection/tracking algorithm
Obstacle detection
Bounding box and class of obstacle in image
Obstacle localization
Calculation of relative position in water-plane
Obstacle tracking
Tracking obstacle using tracking filter
Fusion of multiple tracking results
Tracking 1 (sensor 1)
System Tracks (fused data)
Tracking 2 (sensor 2)
Prediction
Fusion of tracking results using sensor fusion technique
Real sea area application and verification
Program

Figure 2. Process of the proposed recognition system

2. Camera-based marine obstacles detection

An accurate obstacle detection algorithm is the most important element of camera recognition technology. It is the most precedent process among the three steps (detection, location estimation, and tracking) which constitute cognitive technology. Because, if accurate detection is possible, relatively accurate tracking is possible even with a simple tracking algorithm. However, if the detection error is large, accurate tracking is impossible with any algorithm.

In this paper, an obstacle detection algorithm based on deep learning was constructed for accurate detection. A bounding box surrounding an obstacle was extracted from a camera image, and the relative position of the obstacle was calculated using the bounding box and the posture of the camera. In this process, we found that most of the algorithms designed for real-time obstacle detection had a limitation of low accuracy. To overcome this, we introduced an attention module.

2.1. Object detection algorithm

For image-based obstacle detection, this paper used a deep learning algorithm that is rapidly developing. Object detection in images using CNN has been studied in various ways. Based on the structure of the object detection algorithm, it can be largely divided into a one-stage algorithm and a two-stage algorithm. Representatively, there is the one-stage algorithm YOLO series algorithm and the two-stage algorithm Regions with CNN features (R-CNN) series algorithm.

The one-stage algorithm is an algorithm that calculates regression and classification in one step based on features extracted from images. Because it uses the features once, it has the advantage of fast computation speed. However, there is a limitation in that its detection accuracy is low than the common two-stage algorithm. On the other hand, in the two-stage algorithm, regression and classification are performed in two steps. First, the proposed region is extracted through the Region Proposal Network (RPN), and regression and classification are performed. So it has the characteristic that the computation speed is relatively slow. However, a two-stage algorithm is used when high detection accuracy is required regardless of speed.

This paper selected a one-stage algorithm with a relatively fast computation speed, focusing on application to actual USV. An obstacle detection algorithm was constructed based on the YOLOv5 algorithm. The structure of the YOLOv5 algorithm is shown in Figure 3. The YOLOv5 algorithm has nano, small, medium, large, and x-large versions that have the same structure and differ only in the depth and width of the network. Table 2 shows each of the five algorithms' detection accuracy and computation speed. As a result, because the specification of the PC to be loaded into the USV is relatively low, the calculation speed is slower than that of a general pc, and the trade-off between computation speed and detection accuracy, YOLOv5m can guarantee the maximum computation time and minimum accuracy required. Therefore, in this paper, an obstacle detection algorithm was constructed based on the YOLOv5m algorithm.

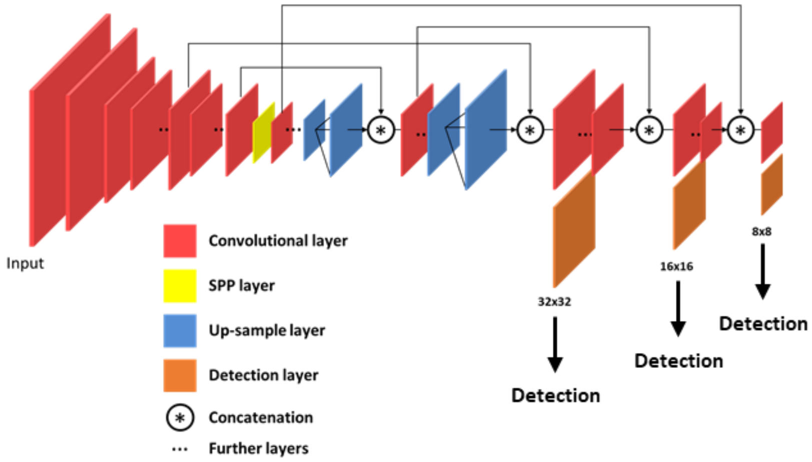


Figure 3. Structure of YOLOv5

Table 2. Specification of YOLOv5 trained with COCO dataset

Algorithm	**mAP^{val}** **0.5:0.95 (%)**	**mAP^{val}** **0.5 (%)**	**Speed** **V100 (ms)**	**Params** **(M)**
YOLOv5n	28.0	45.7	6.3	1.9
YOLOv5s	37.4	56.8	6.4	7.2
YOLOv5m	45.4	64.1	8.2	21.2
YOLOv5l	49.0	67.3	10.1	46.5
YOLOv5x	50.7	68.9	12.1	86.7

YOLO-based algorithms have the advantage of fast computation speed due to the characteristic of one-stage algorithms. However, they also have the disadvantage of low detection accuracy compared to two-stage algorithms. Therefore, in this paper, we improved the accuracy by inserting CBAM [3] before each detection layer of the YOLOv5m algorithm.

CBAM is an attention module that functions as a layer trained to calculate appropriate weights according to input values. It compensates for the decrease in accuracy when an input not learned in the algorithm comes in. CBAM is applied to the obstacle detection algorithm of this paper because it corresponds to mixed attention that considers both channel attention and spatial attention among attention modules. The obstacle detection algorithm applying CBAM to the YOLOv5m algorithm is shown in Figure 4.

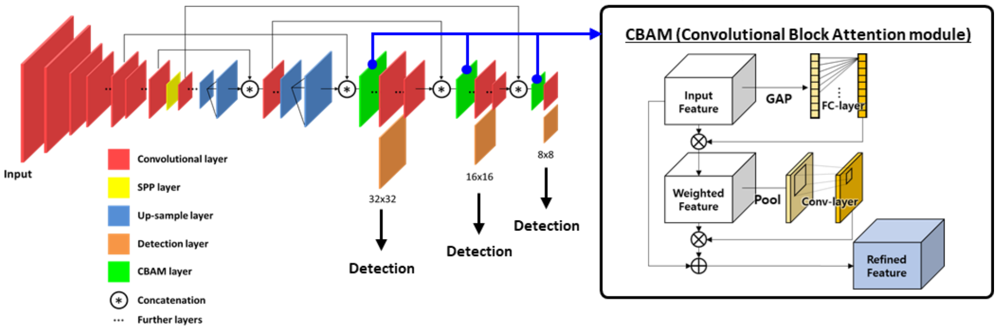


Figure 4. Structure of obstacle detection algorithm

To train the obstacle detection algorithm, a data set was constructed based on images acquired in Changwon, Pyeongtaek, and Jebudo Islands in the Republic of Korea. It is classified into a data set for obstacle detection in EO images and a data set for detection in IR images. Each data set comprises training, verification, and test data sets. First, the EO detection algorithm was trained and analyzed with 4,641 training data, 516 verification data, and 553 test data. For the IR detection algorithm, since the lack of IR data, the data was augmented through image flipping. Training and analysis were performed with 2,952 training data, 328 verification data, and 122 test data. All data was captured by a camera (1-channel EO, 1-channel IR, 3-channel EO) mounted on a vessel with a length of 8 to 12 meters. An example of image data is shown in Figure 5 below.

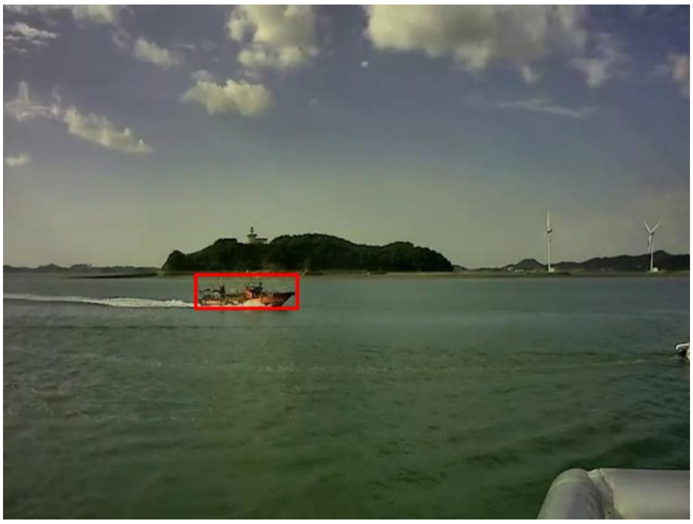
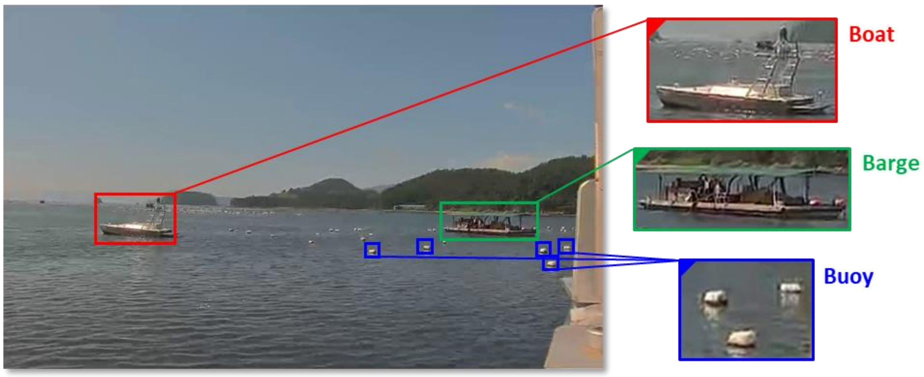
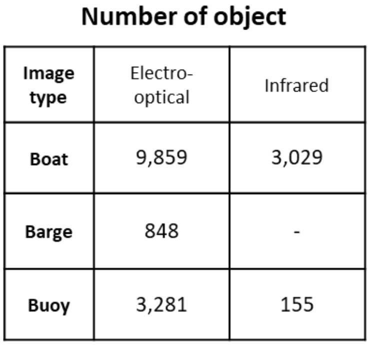


Figure 5. Sample of the image data

The maritime obstacles were classified into three classes, and the types are shown in Figure 6 below. The ‘Boat’ class corresponds to a general type of boat or USV and is one of the main perceived obstacles at the coast because it usually moves at relatively high speeds. Therefore, the largest number of objects among the total object to training in both EO and IR detection algorithms were assigned to this class. The ‘Barge’ class includes fixed objects such as aquaculture auxiliary vessels. Because it is fixed at sea, the recognition priority is relatively low. Moreover, there are many cases where they have external differences from general boats, so we separate classes to increase the detection accuracy of the main recognition target, ‘Boat’. The least number of objects is included in the optical data, and the thermal image data was included in the ‘Boat’ for learning because the entire data was insufficient. Finally, the ‘Buoy’ class includes all water surface markers regardless of size and shape, and the second largest number of objects were used for learning.





Number of object

Image type	Electro-optical	Infrared
Boat	9,859	3,029
Barge	848	-
Buoy	3,281	155

Figure 6. Classification of maritime obstacles

The obstacle detection algorithm was trained on an INTEL i7-10700, NVIDIA GeForce RTX 3080 Ti, 32GB RAM, and Microsoft Windows 10 environment. The EO detection algorithm was trained with batch size 8 and 200 epochs, and the thermal image detection algorithm was trained with batch size 8 and 150 epochs.

2.2. Position transformation

In order to track an obstacle in a 3D space based on the detection, the detected bounding box must be converted into 3D location information (distance, bearing). The method of converting the bounding box information detected in the image into 3D spatial coordinates for obstacle tracking is largely divided into a method using a stereo camera and a method using a monocular camera.

First, the stereo camera method uses the parallax and disparity of two parallel cameras, like the human eyes, as a visual clue to estimate the position through triangulation. It has the advantage of changing the position of all points that match each other on the two image planes. However, there are several difficulties in setting, such as additional calibration between the two cameras.

On the other hand, the monocular camera method uses the horizon appearing in the image as a most important visual clue. Based on the pin-hole camera model, we transform the position into a three-dimensional space using the distance between the horizon line and the point where the obstacle contacts the water surface (pixel) and the distance between the principal point of the image and the obstacle (pixel). Since the visual clue is a single horizon line, there is a limit that only the point in contact with the water surface can be transformed. However, there is an advantage in that the position can be transformed without additional settings.

In this paper, to apply the research results to a camera mounted on a USV where rapid motion frequently occurs, a position transformation method using a monocular camera with relatively fewer setting errors during the operation was adopted. In addition, since most of the maritime obstacles targeted for detection and tracking are in contact with the water surface, the limitations of the position transformation method adopted must be addressed.

Position transformation is largely divided into obtaining relative bearing and relative distance of the detected obstacle. The process of calculating the relative bearing is shown in Figure 7. First, an obstacle position vector is defined at the point where the obstacle detected on the camera coordinate system contacts the water surface. Afterward, the camera coordinate is transformed from the camera coordinate to the body-fixed coordinate, considering the location and angle of the camera installed on the boat. For example, in the case of a panoramic camera (3 channels), the images taken by multiple cameras are concatenated. At this time, each camera is rotated in reverse as much as the installation angle. Finally, the obstacle vector defined in the body-fixed coordinate is converted to the global coordinate, considering the posture of own boat. Through the transformation process above, the direction of the obstacle can be calculated from the heading of boat on the global coordinate. In the whole transformation process, the quaternion was used to rotate so that the gimbal-lock problem did not occur.

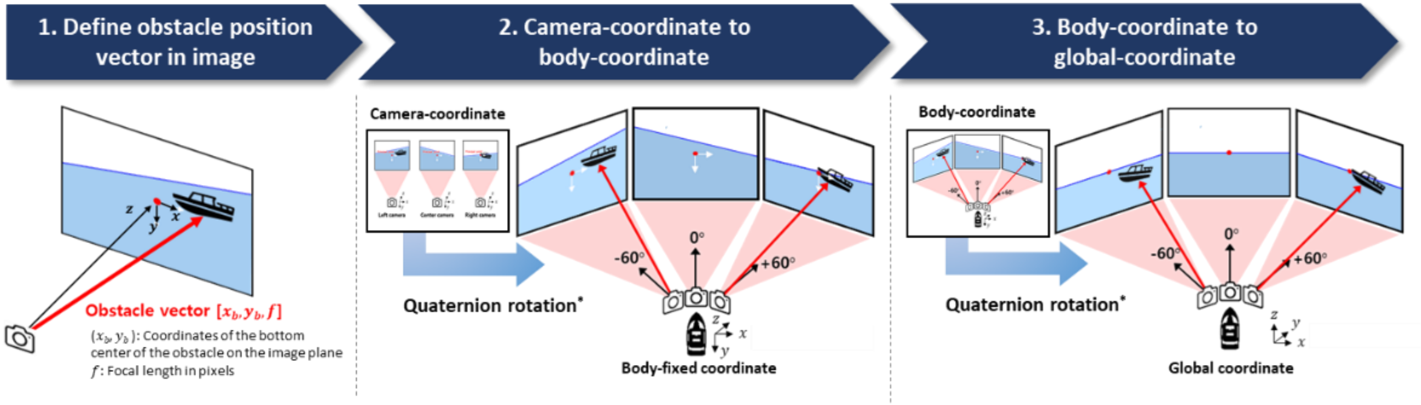

1. Define obstacle position vector in image
2. Camera-coordinate to body-coordinate
3. Body-coordinate to global-coordinate
z
x
y
Obstacle vector $[x_b, y_b, f]$
(x_b, y_b): Coordinates of the bottom center of the obstacle on the image plane
f: Focal length in pixels
Camera-coordinate
Left camera
Center camera
Right camera
Quaternion rotation*
-60°
0°
+60°
z
x
y
Body-fixed coordinate
Body-coordinate
Quaternion rotation*
-60°
0°
+60°
z
y
x
Global coordinate

Figure 7. Process of the relative bearing calculation

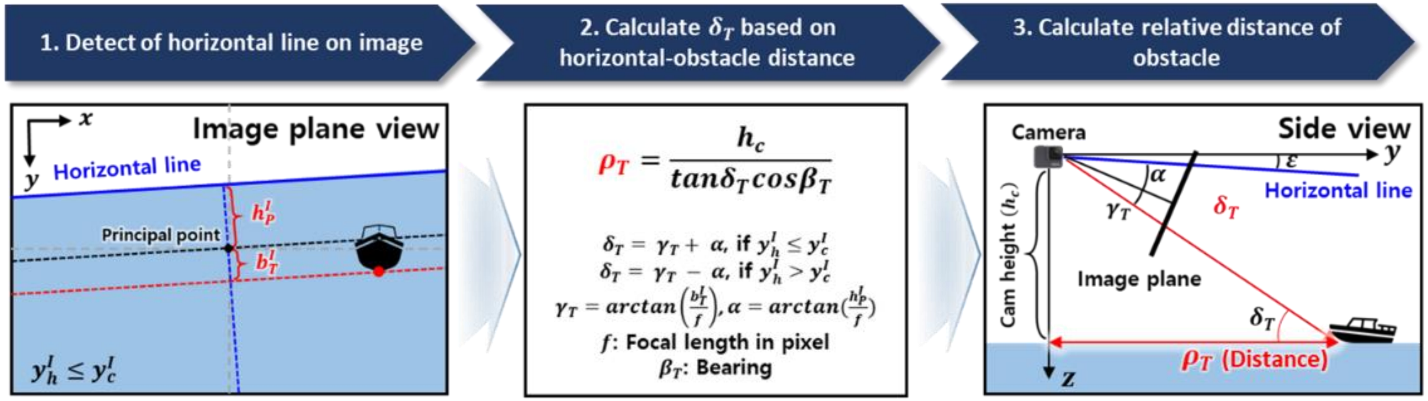

1. Detect of horizontal line on image
2. Calculate δ_T based on horizontal-obstacle distance
3. Calculate relative distance of obstacle
Image plane view
x
y
Horizontal line
h_P^I
Principal point
b_T^I
$y_h^I \leq y_c^I$
$\rho_T = \frac{h_c}{tan\delta_T cos\beta_T}$
$\delta_T = \gamma_T + \alpha$, if $y_h^I \leq y_c^I$
$\delta_T = \gamma_T - \alpha$, if $y_h^I > y_c^I$
$\gamma_T = arctan\left(\frac{b_T^I}{f}\right), \alpha = arctan(\frac{h_P^I}{f})$
f: Focal length in pixel
β_T: Bearing
Camera
Side view
ε
y
α
γ_T
δ_T
Horizontal line
Cam height (h_c)
Image plane
δ_T
ρ_T (Distance)
Z

Figure 8. Process of the relative distance calculation

The process of calculating the relative distance of the obstacle is shown in Figure 8. First, a horizontal line was detected, an important visual clue for distance calculation in the monocular method. In this paper, the horizontal line on the image is detected based on the posture measured from the gyro sensor mounted on the USV so that the horizontal line detection process is not affected by the quality of the image or the surrounding environment for accurate distance calculation.

Using the pixel distance between the detected horizontal line and the obstacle bounding box, the angle δ_T between a straight line parallel to the water surface and a straight line connecting the camera and the obstacle can be calculated. Finally, through the following Eq. (1), the relative distance of the obstacle can be obtained using δ_T and the camera installation height.

$$\begin{aligned} \rho_T &= \frac{h_c}{\tan(\delta_T)\cos(\beta_T)} \\ \delta_T &= \gamma_T + \alpha, \ \text{if}\, y_h^I \le y_c^I \\ \delta_T &= \gamma_T - \alpha, \ \text{if}\, y_h^I > y_c^I \end{aligned} \tag{1}$$

f represents the focal length of the camera in pixels unit, and β_T represents the bearing of the obstacle with respect to the camera heading. γ_T is the angle calculated by the straight line connecting the camera origin and the principal point of the image plane with the straight line connecting the camera origin and the obstacle, calculated as $\arctan(\frac{b_T^I}{f})$. α is the angle calculated by the straight line connecting the camera origin and the horizontal line with the straight line connecting the camera origin and the principal point of the image plane, calculated as $\arctan(\frac{h_P^I}{f})$.

3. Marine obstacles tracking

In obstacle tracking, we estimate a motion data of the obstacle based on the positional information (distance and bearing) of the obstacle acquired as a result of the detection. The motion information of the obstacle to be estimated in this paper is the trajectory, Course Over Ground (COG), and Speed Over Ground (SOG). The obstacle tracking process is divided into data association which is matching tracking data up to the previous time and detection information of the current time, and tracking, which is estimating motion data based on matched detection information.

3.1. Data association

Data association is a method of matching tracking data up to a previous point in time with detection information of a current point. And there are various methods, such as a location-based matching method or a visual feature-based matching method. There is no significant difference when the obstacles are spread out on the sea according to the matching method. However, in an environment where obstacles are dense, there are limitations to the location-based matching method. A small error in the detection step can increase the size of the error in the location estimation process. Therefore, we used a matching method based on visual features.

Bewley et al. [11] proposed Simple Online Realtime Tracking (SORT), a representative data association algorithm in computer vision. As shown in Figure 9, the SORT algorithm first predicts the motion of the bounding box which is surrounding the object on the image plane using the Kalman Filter (KF). Then, the intersection over union (IoU) between the predicted value and the detected value is defined as the similarity (reverse of cost). Based on the similarity, the detection information is assigned to each tracking data through the Hungarian allocation algorithm. The SORT algorithm has the advantage that real-time association is possible. However, it has the problem that the object motion predicted using KF is only the motion on the image plane and does not reflect the object motion in the actual 3D space.

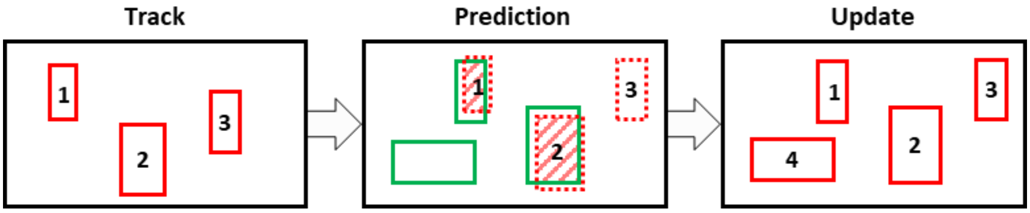


Figure 9. Process of the SORT algorithm

In this paper, we proposed a data association algorithm that reflects motion data estimated through obstacle tracking to enable robust association even if multiple obstacles overlap or are covered in a situation where many obstacles are concentrated in the actual sea. The proposed data association algorithm significantly improved two things from the SORT-based algorithm. First, it is a prediction method for the obstacle bounding box of the next time step $(t+1)$. When the motion data of the obstacle estimated through tracking up to the previous time step $(t-1)$ is defined as a trajectory $(\hat{x}_{t-1}, \hat{y}_{t-1})$, COG $(\hat{\phi}_{t-1})$, and SOG $(\hat{v}_{t-1})$, the obstacle position $(\hat{x}'_t, \hat{y}'_t)$ at the current time step (t) is the derived as following Eq. (2).

$$\begin{aligned} \hat{x}'_t &= \hat{x}_{t-1} + \hat{v}_{t-1}\sin(\hat{\phi}_{t-1})\Delta t \\ \hat{y}'_t &= \hat{y}_{t-1} + \hat{v}_{t-1}\cos(\hat{\phi}_{t-1})\Delta t \end{aligned} \tag{2}$$

In order to estimate the bounding box through the calculated $\hat{x}'_t$ and $\hat{y}'_t$, the inverse transformation process of the position transformation algorithm used in the previous obstacle position estimation process was used. First, at the time (t), if we define (x_o, y_o), as the position of the own boat, ϕ_h as the heading of the own boat, ϕ_r as the roll of the own boat, h_c as the camera installation height, and f as the camera focal length, the numerical value for position transformation can be calculated as shown in Eq. (3).

$$
\begin{aligned}
\rho &= \sqrt{(x_o - \hat{x}_t')^2 + (y_o - \hat{y}_t')^2} \\
\beta &= \arctan(\frac{\hat{x}_t' - x_o}{\hat{y}_t' - y_o}) - \phi_h \\
\beta_T^I &= \frac{\tan(\beta) \times f}{\cos(\phi_r)} \\
\delta_T &= \arctan(\frac{h_c}{d \times \cos(\beta)}) \\
b_T^I &= f \times \tan(\delta_T - \arctan(\frac{h_h \times \cos(\phi_r)}{f}))
\end{aligned}
\tag{3}
$$

ρ is the distance between the ship and the obstacle [m], β represents the bearing of the obstacle based on the heading of the ship, and can be calculated as above. Based on the calculated values, the image coordinates (x_i, y_i) corresponding to the global coordinates $(\hat{x}'_t, \hat{y}'_t)$ of the obstacle in the image with width w_i and height h_i are shown as Eq. (4).

$$
\begin{aligned}
x_i &= \frac{w_i}{2} - b_T^I \times \sin(\phi_r) + \beta_T^I \times \cos(\phi_r) \\
y_i &= \frac{h_i}{2} + b_T^I \times \cos(\phi_r) + \beta_T^I \times \sin(\phi_r)
\end{aligned}
\tag{4}
$$

An example of applying the bounding box prediction method at the next time step is shown in Figure 10.

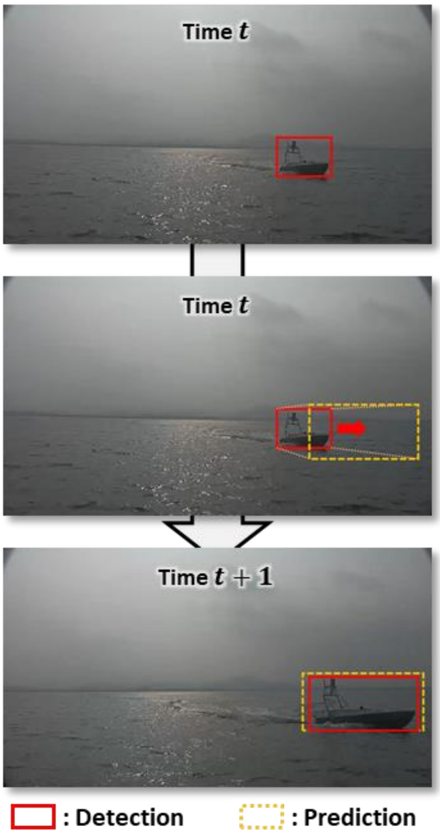


Figure 10. Example of the proposed algorithm for data association

The second improvement is to modify the cost of the Hungarian allocation algorithm to suit this paper. IoU, used as the cost of the Hungarian allocation algorithm in the proposed SORT algorithm, can evaluate the similarity based on the size and location of bounding boxes together. However, when the camera is not fixed and moving, a bounding box shift on the image often occurs. Accordingly, IoU alone has a limitation in that proper similarity cannot be evaluated.

Therefore, in this paper, the center distance between the bounding boxes was additionally applied to the cost to calculate the appropriate cost even when the bounding box shift occurred. The center distance between the bounding boxes means the positional similarity of the two bounding boxes. Positional similarity is the most common similarity used in various matching algorithms, including the traditional Nearest Neighbor (NN) algorithm. Therefore, by adding the positional similarity in the image to the cost, the similarity that IoU cannot calculate can be compensated.

However, defining only the center distance of the bounding box as the cost has a disadvantage. It is very likely to make an error when many obstacles are dense. In this paper, the cost is defined based on the center distance of the bounding box but multiplied by the IoU cost so that IoU plays a dominant cost in matching when obstacles are dense. Conversely, when the bounding box shift occurs, the bounding box center distance is designed to play a dominant cost in matching. The defined equation is shown in the following Eq. (5).

$$\begin{gathered}\text{cost=}\sqrt{(\Delta x_{p\_d})^2+(\Delta y_{p\_d})^2}\times(1-IoU(bbox_p,bbox_d)) \\ IoU(bbox_p,bbox_d)=\frac{bbox_p\bigcap bbox_d}{bbox_p\bigcup bbox_d}\end{gathered} \tag{5}$$

The cost of the Hungarian allocation algorithm was defined as a product of the central distance between the bounding boxes and the IoU cost. $bbox_p, bbox_d$ mean the predicted bounding box and the detected bounding box, respectively. Bounding box information is consisted of the $[x_1 \quad y_1 \quad x_2 \quad y_2]$, which is the top-left and bottom-right coordinate of the surrounding box. And $\Delta x_{p\_d}, \Delta y_{p\_d}$ respectively represent the central distance in a width direction and in a height direction between the predicted bounding box and the detected bounding box. The function IoU on Eq. (5) is a function that calculates the IoU of two bounding boxes.

3.2. Tracking filter

In this paper, an adaptive tracking filter suitable for a detection method using a camera is designed to estimate the motion data of the obstacle, such as trajectory, COG, and SOG. Motion is estimated using distance and bearing, which are the location information of the obstacle. The proposed adaptive tracking filter is designed based on EKF (Kim and Park [12]), one of the recursive filters for estimating nonlinear systems based on sensor measurements. The basic structure of EKF is shown in Figure 11.

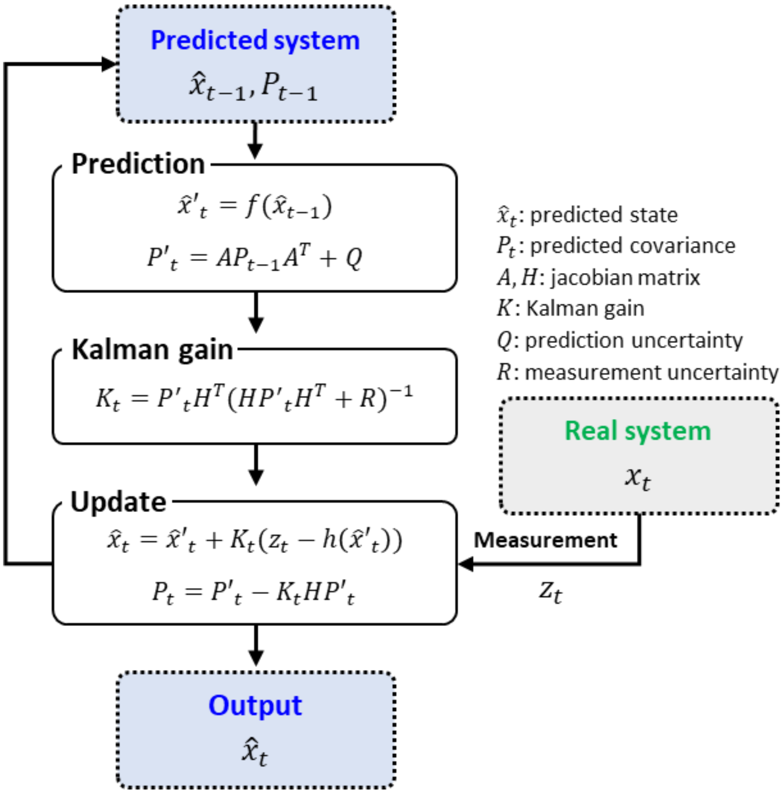

Predicted system
$\hat{x}_{t-1}, P_{t-1}$
Prediction
$\hat{x}'_t = f(\hat{x}_{t-1})$
$P'_t = AP_{t-1}A^T + Q$
Kalman gain
$K_t = P'_t H^T(HP'_t H^T + R)^{-1}$
Update
$\hat{x}_t = \hat{x}'_t + K_t(z_t - h(\hat{x}'_t))$
$P_t = P'_t - K_t HP'_t$
Output
$\hat{x}_t$
$\hat{x}_t$: predicted state
P_t: predicted covariance
A, H: jacobian matrix
K: Kalman gain
Q: prediction uncertainty
R: measurement uncertainty
Real system
x_t
Measurement
z_t

Figure 11. Structure of the Extended Kalman Filter

In this paper, the state vector is defined as $[x, y, v, \phi]$, x and y are absolute positions, ϕ is COG, and v is SOG of the tracked obstacle. The sensor measurement vector is $[\rho, \beta]$, where ρ is the distance from the own boat and β is the bearing from the heading of the own boat.

The system model of the obstacle was assumed as constant velocity motion, as shown in Eq. (6). Unlike land obstacles or land vehicles, marine obstacles are generally characterized by low acceleration. Moreover, the tracking period of the detection method proposed in this paper is sufficiently short, less than 0.1 second per tracking update, so the system model assumption is suitable.

The correlation between the state and measurement vectors was defined as Jacobian matrices A and H. In this paper, matrices A and H are defined as Eq. (7). In the equation, x_o and y_o mean the absolute position of the boat.

$$\hat{x}'_t = \hat{x}_{t-1} + \hat{v}_{t-1}\Delta t \tag{6}$$

$$\begin{gathered}
A = \begin{bmatrix} 1 & 0 & \sin(\phi)dt & v\cos(\phi)dt \\ 0 & 1 & \cos(\phi)dt & -v\sin(\phi)dt \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \\
H = \begin{bmatrix} \dfrac{x-x_o}{\sqrt{(x-x_o)^2+(y-y_o)^2}} & \dfrac{y-y_o}{\sqrt{(x-x_o)^2+(y-y_o)^2}} & 0 & 0 \\ -\dfrac{(y-y_o)}{(x-x_o)^2+(y-y_o)^2} & \dfrac{(x-x_o)}{(x-x_o)^2+(y-y_o)^2} & 0 & 0 \end{bmatrix}
\end{gathered} \tag{7}$$

In most tracking-related studies using the Kalman filter, when setting the measurement error covariance, a fixed value obtained by multiplying the error covariance of the sensor by an appropriate margin is used. However, in the case of detecting an obstacle using a camera with the proposed detection algorithm, an obstacle detected from a close location has a low error variability, and an obstacle detected from a distant location has a high error variability. Since the variability of sensor measurement that varies depending on the detection location cannot be defined as one common error covariance, in this paper, the variability according to the detection location is reflected in the tracking by defining the adaptive error covariance.

In this paper, since the variability of sensor measurement is caused by the maritime obstacle detection algorithm, the error covariance of the sensor can be estimated through the distribution of detection information. Therefore, the distribution of bounding boxes of obstacles detected in images (consecutive frames) was first analyzed to estimate the variability arising from the obstacle detection algorithm. Then, to estimate the measurement distribution from the distribution of the bounding box, the variability of the sensor measurement by the obstacle detection algorithm was finally estimated by calculating the change in the relative distance of the obstacle as the bounding box fluctuated by 1 pixel.

As shown in Figure 12 below, we find that the standard deviation of the bounding box distribution by the obstacle detection algorithm is 1.82 pixels. Also, as the bounding box is distributed by 1 pixel, the distribution of the relative distance can be calculated as the following Eq. (8) by differentiating Eq. (3).

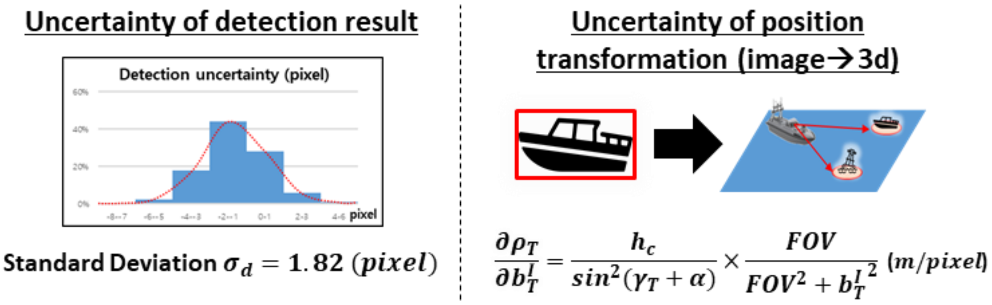

Uncertainty of detection result
Detection uncertainty (pixel)
60%
40%
20%
0%
-8~-7
-6~-5
-4~-3
-2~-1
0~1
2~3
4~6
pixel
Standard Deviation $\sigma_d = 1.82\ (pixel)$
Uncertainty of position transformation (image→3d)
$\frac{\partial \rho_T}{\partial b_T^I} = \frac{h_c}{sin^2(\gamma_T + \alpha)} \times \frac{FOV}{FOV^2 + {b_T^I}^2}\ (m/pixel)$

Figure 12. Uncertainty of obstacle detection algorithm

$$\begin{gathered}\sigma_d = 1.82 \\ \frac{\partial \rho_T}{\partial b_T^I} = \frac{h_c}{\sin^2(\gamma_T + \alpha)} \times \frac{FOV}{FOV^2 + (b_T^I)^2}\end{gathered} \tag{8}$$

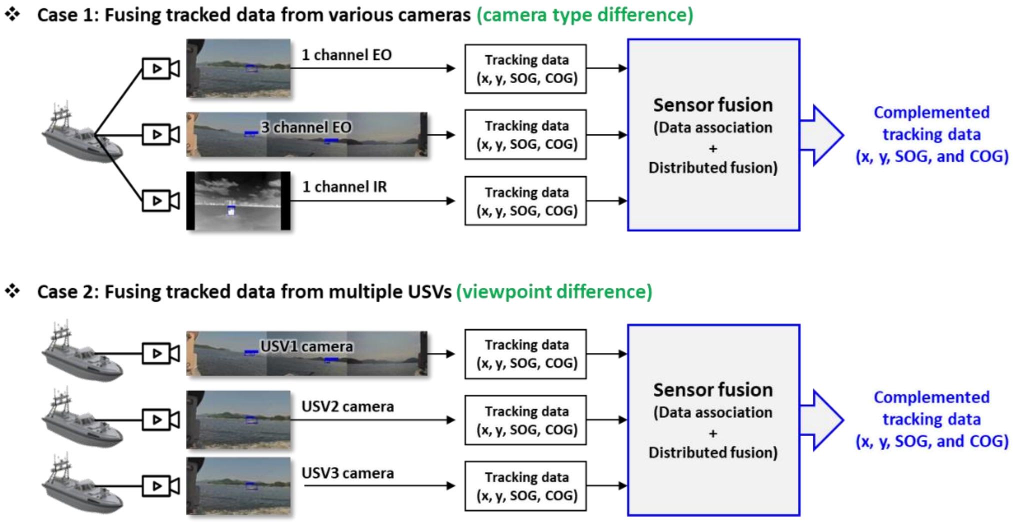
FOV means the horizontal field of view of the camera in radian, and other parameters are shown in Eq. (3).

The adaptive error covariance defined by reflecting the distribution of the obstacle detection algorithm and introducing a moving average filter for smoothing is shown in Eq. (9). R is the adaptive error covariance, σ_ρ is the standard deviation of the distance error, and σ_β is the standard deviation of the bearing error. Because the standard deviation of the bearing error distribution is sufficiently small compared to the distance error, we defined it as a fixed value.

$$
\begin{aligned}
\sigma_\rho &= moving\ \ average(\sigma_d \times \frac{\partial \rho_T}{\partial b_T^I}) \\
\sigma_\beta &= 0.03 \\
R &= \begin{bmatrix} (\sigma_\rho)^2 & 0 \\ 0 & (\sigma_\beta)^2 \end{bmatrix}
\end{aligned}
\tag{9}
$$

4. Sensor fusion

Fusion between multiple tracked data on the same obstacle was performed to increase tracking accuracy and reliability in various situations. We adopted fusion in two ways: fusing tracked data from three types of cameras (1-channel EO, 1-channel IR, 3-channel EO) mounted on one boat and fusing tracked data from each boat observing obstacles at various viewpoints at the same time. A graphical explanation of Each case is shown in Figure 13.



❖ Case 1: Fusing tracked data from various cameras (camera type difference)
1 channel EO
3 channel EO
1 channel IR
Tracking data (x, y, SOG, COG)
Tracking data (x, y, SOG, COG)
Tracking data (x, y, SOG, COG)
Sensor fusion (Data association + Distributed fusion)
Complemented tracking data (x, y, SOG, and COG)
❖ Case 2: Fusing tracked data from multiple USVs (viewpoint difference)
USV1 camera
USV2 camera
USV3 camera
Tracking data (x, y, SOG, COG)
Tracking data (x, y, SOG, COG)
Tracking data (x, y, SOG, COG)
Sensor fusion (Data association + Distributed fusion)
Complemented tracking data (x, y, SOG, and COG)

Figure 13. Scenarios of the sensor fusion case

In order to fuse tracked data from different cameras, a process of matching data indicating the same obstacle between different tracking data must be preceded. It is called data association. For example, when each of several cameras tracks an obstacle, it is associated if tracking for the same obstacle is performed by more than one camera.

In this paper, the association between different tracking data was performed using the Nearest Neighbor (NN) algorithm, which is most commonly used for data association. The NN algorithm judges the closest tracking data within the threshold as the data from the same obstacle based on the location of the tracking data, and the formula is shown in Eq. (10). x and y mean the absolute coordinates of the tracking data, and subscript i and j represent i^{th} camera and j^{th} camera, respectively.

$$\begin{gathered} \text{True,} \quad \text{if } \sqrt{(x_i - x_j)^2 + (y_i - y_j)^2} \le threshold \\ \text{False,} \quad \text{if } \sqrt{(x_i - x_j)^2 + (y_i - y_j)^2} > threshold \\ i, j \in camera\ set \end{gathered} \tag{10}$$

The method of fusing data between associated tracking data can be largely classified into sensor-to-global fusion and sensor-to-sensor fusion, as shown in Figure 14. First, sensor-to-global fusion is a method of fusing the tracking data to be fused with the system track by defining a system track (global track) that has an additional tracking process. On the other hand, sensor-to-sensor fusion is a method that does not define a separate system track but fuses matched tracking data at every moment. When defining a system track, it goes through an additional tracking process, so there is a smoothing effect, but there is a disadvantage that error accumulation may occur accordingly.

In this paper, the sensor-to-sensor fusion method was adopted to maintain the tracking characteristics of each camera during the sensor fusion and to eliminate the accumulation of errors due to an additional tracking process. While the sensor fusion process, when tracking data for fusion was missing, fused by predicting tracking data at that time based on the current tracking data.

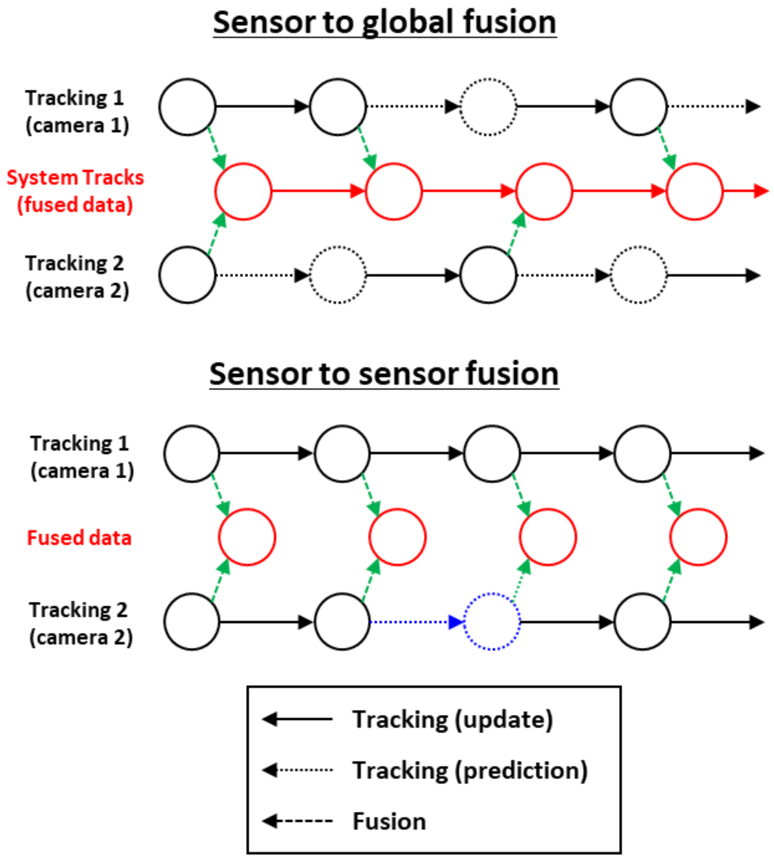


Figure 14. Type of fusion process

The sensor fusion algorithm used in this paper is fast Covariance Intersection (fast CI) (Fränken and Hüpper [13]). The original CI algorithm is a weighted fusion algorithm that finds the weights that minimize the trace or determinant of the resulting error covariance and fuses using the found weights. (see Eq. (11)) And also, it is a candidate that yields consistent estimates independent of network structure and any possible cross-correlation between local estimates. (Fränken and Hüpper [13])

$$\begin{aligned} \hat{x}_{sf} &= P_{sf}[wP_1^{-1}\hat{x}_1 + (1-w)P_2^{-1}\hat{x}_2] \\ P_{sf} &= [wP_1^{-1} + (1-w)P_2^{-1}]^{-1} \\ w = \arg\min(&trace(P_{sf})) \,\|\, \arg\min(\det(P_{sf})) \end{aligned} \tag{11}$$

However, the original CI algorithm has a disadvantage because it takes a long time to calculate due to the iterative process of finding the appropriate weights. Therefore, in this paper, we fuse tracking data using a fast CI algorithm designed to enable real-time calculation by replacing nonlinear optimization to find weights with numerical calculation. Since there is a possibility for more than two tracking data to be fused (more than three cameras), the fast CI algorithm for fusion between two or more sensor data was used. The detailed equation to calculate the weights, resulting state, and resulting covariance of the state is shown in Eq. (12). (Mitchell [14])

$$\begin{aligned}
\hat{x}_{sf} &= \sum_m w_m P_{sf} P_m \hat{x}_m \\
P_{sf} &= [\sum_m w_m P_m^{-1}]^{-1} \\
w_m &= \frac{\left|S\right| - \left|S - P_m^{-1}\right| + \left|P_m^{-1}\right|}{M\left|S\right| + \sum_m (\left|P_m^{-1}\right| - \left|S - P_m^{-1}\right|)} \\
S &= \sum_m P_m^{-1}
\end{aligned} \tag{12}$$

5. Applications

The application was conducted based on navigation data acquired from the coast for three years (2020 ~ 2022, Changwon, Pyeongtaek, and Jebudo Islands in Republic of Korea). The accuracy of detection, tracking, and fusion algorithms were analyzed on three trial tests. The graphical description of each case and the vessel specifications are shown in Figure 15. Regarding the obstacle tracking result, the tracking result within 18m, corresponding to about 80% accuracy based on the general GPS error of 15m, was evaluated as a meaningful tracking result.

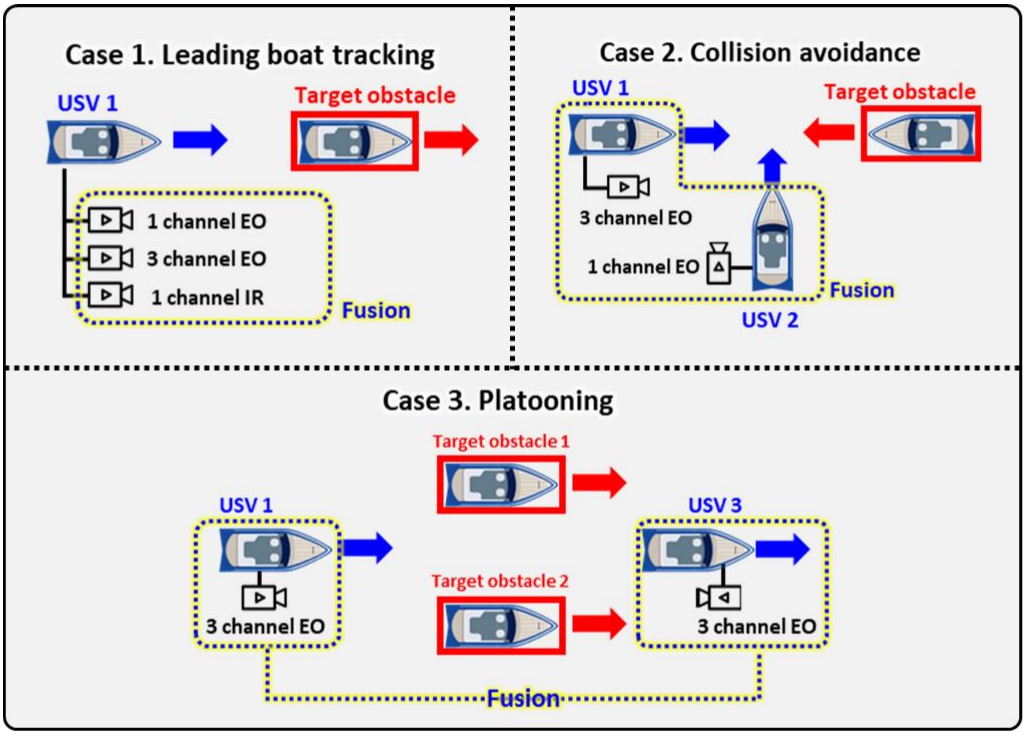


Figure 15. Application scenarios

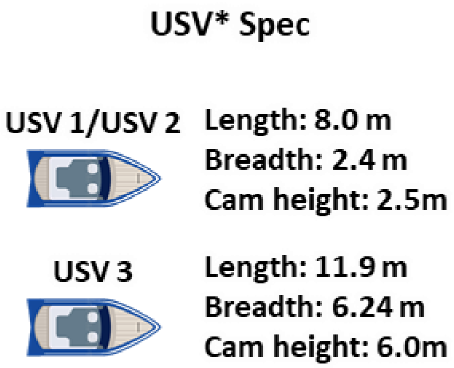


Figure 16. Specification of the USVs

Case 1 is an example of detecting and tracking by defining the leading boat as the target obstacle in chasing the boat ahead. USV 1 is equipped with three types of cameras (1-channel EO, 1-channel IR, 3-channel EO), and the data tracked by each camera are fused together. Through this case, the differences in tracking results from different types of cameras can be confirmed, and the effects of error reduction and disadvantage compensation through fusion can be confirmed.

Case 2 is an example in which USV 1 and USV 2 detect and track by defining a boat approaching from the right as a target obstacle based on Figure 15 in a situation where three boats assume collision and avoidance. The data tracking the target obstacle simultaneously from the viewpoints of USV 1 and USV 2 was fused with each other. Through this case, it can be confirmed that the error characteristics of the distance error and bearing error of the detection method using the camera appear differently depending on the viewpoint, and the error reduction due to fusion can be confirmed.

Case 3 is an example in which USV 1 and USV 2 detect and track by defining target obstacles 1 and 2, respectively, for the two boats located in the middle where four boats operate in a cluster (see Figure 15). Similar to Case 2, the data tracked simultaneously from the viewpoints of USV 1 and USV 2 was fused with each other. Through this case, the occurrence of tracking error according to the turning radius of the obstacle was confirmed, and the effect of fusion was analyzed when there was no significant difference between the two tracking results.

5.1. Obstacle detection

The accuracy of the maritime obstacle detection algorithm was calculated based on Average Precision (AP). The EO detection algorithm and the IR detection algorithm were trained separately, and the data for accuracy analysis were 553 EO images and 122 EO images. The environment for computing speed measurement is Intel® Core™ i7-10700, GeForce GTX 3080 Ti, 32GB RAM, and Microsoft Windows 10. The accuracy analysis results are shown in Table 3 below.

Table 3. Accuracy of obstacle detection algorithm

Detection algorithm	**Detection image**	**AP (Average Precision)**	**Computation time (sec/frame)**	**Computation speed (FPS)**
YOLOv5m	Electro Optical (EO)	95.75%	0.0266	37.6
v5m + CBAM (proposed)	Electro Optical (EO)	95.98%	0.0284	35.2
YOLOv5m	Infrared (IR)	94.44%	0.0266	37.6
v5m + CBAM (proposed)	Infrared (IR)	95.67%	0.0284	35.2

Compared to the IR detection algorithm, the training data was abundant in the case of the EO detection algorithm, so the effect of introducing CBAM was not very noticeable. However, it was confirmed that the IR detection algorithm, which increased the training data through data augmentation (such as image flipping) due to a relatively small dataset, showed a significant effect of introducing CBAM, with about 1.23% accuracy improvement. In addition, while the obstacle detection accuracy increased by introducing CBAM, the computation time per frame increased by a very small amount from 0.0266sec to 0.0284sec. Nevertheless, the detection accuracy of both obstacle detection algorithms is over 90%, which is not enough to perform tracking based on image detection results.

5.2. Obstacle tracking and sensor fusion

Analysis of obstacle tracking and fusion results was performed for the three cases described above. First, the tracking results individually in each camera were analyzed, and the fusion of data tracked by multiple cameras was analyzed.

5.2.1. Case 1: Leading boat tracking

Case 1 is an example of detecting and tracking by defining the leading boat as the target obstacle in chasing the boat ahead. The distance between the target obstacle and the USV was maintained at about 80 m while tracking, and the target obstacle repeated the turning motion. First, the result of detecting the obstacle in the 3-channel EO video is shown in Figure 17. For 3-channel EO video, the horizontal Field Of View (horizontal FOV) of the camera is 180.0°, and the resolution is 3840×720. The green line in the figure results from detecting a horizontal line on the image based on the posture of USV 1 measured through the gyro sensor. It can be seen that horizontal lines on the image are well detected in all three directional frames.

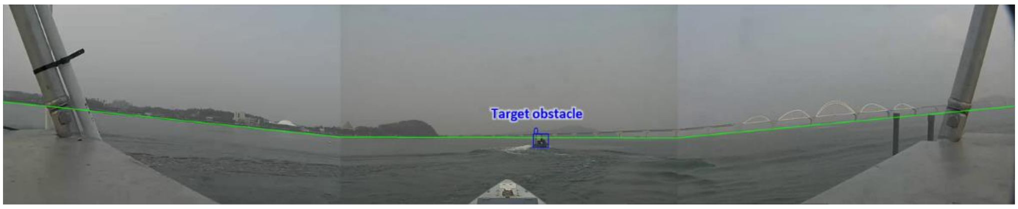


Figure 17. Detection result of target obstacle on 3-channel EO video

Figure 18 shows the result of tracking the trajectory of the obstacle, which is the motion data of the obstacle tracked in the 3-channel EO video. The blue trajectory in the figure means the trajectory of the USV 1, the green trajectory is the ground truth of the target obstacle measured through GPS, and the red trajectory is the trajectory of the target obstacle estimated through the tracking method proposed in this paper.

Although there are parts where some position errors occur more than 18m, the pre-defined benchmark, they are generally similar to ground truth. As a result of calculating the Mean Absolute Error (MAE), the mean error was 6.58m. It is within 80% of the GPS error, and it can be seen that the tracking method proposed in this paper is meaningful.

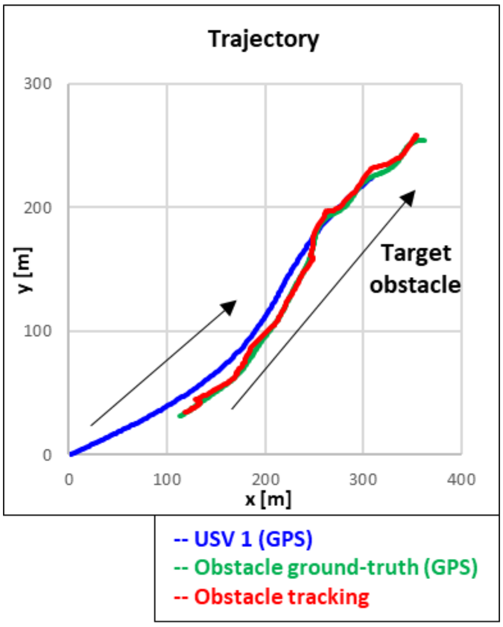


Figure 18. Trajectory tracking result on 3-channel EO video

The part where the relatively large positional error occurred during the tracking process was caused by occlusion due to the wake of the obstacle. As shown in Figure 19, as the obstacle turned rapidly, a strong wake was generated on the water surface, and the part where the obstacle and the water surface came into contact was occluded. Due to the occlusion, a detection error of about 5 pixels was continuously generated compared to ground truth. And it caused a relatively large error when estimating the position of an obstacle.

Detection errors due to occlusion, such as occlusion by wake and occlusion between obstacles, are problems that can frequently occur in the image-based detection process. In the case of errors due to temporary occlusion, they can be removed by the tracking filter. However, this case confirmed that a method to reduce the tracking error is necessary when detection errors due to occlusion occur for a long time.

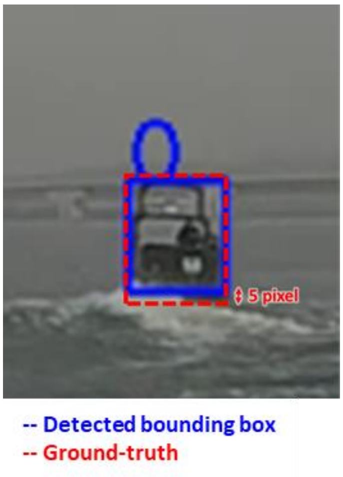


Figure 19. Occluded detection result by wake

The result of tracking the COG and SOG of the obstacle in the 3-channel EO video is shown in Figure 20. As with the previous trajectory, the green graph is the ground truth calculated through GPS, and the red graph is the estimated value through tracking. The units of the drawn graph are degrees and knots, respectively. As a result of calculating the MAE for each result, an error of 12.45° for COG and 1.40 knots for SOG occurred.

Although the tendency was generally similar to ground truth, a relatively large error than the MAE repeatedly occurred in some parts indicated by the arrow in the graph. The indicated parts are sections where the obstacle repeats its turning motion. The error of the parts occurs because the motion of the obstacle is assumed to be a constant velocity motion in the tracking filter proposed in this paper. That is, the tracking delay occurred because the tracking filter could not consider the acceleration due to the rotational motion. Although, if the delay occurred in the corresponding parts is removed, the error is within the average, and the delay is also within 3 seconds, which is a meaningful tracking result.

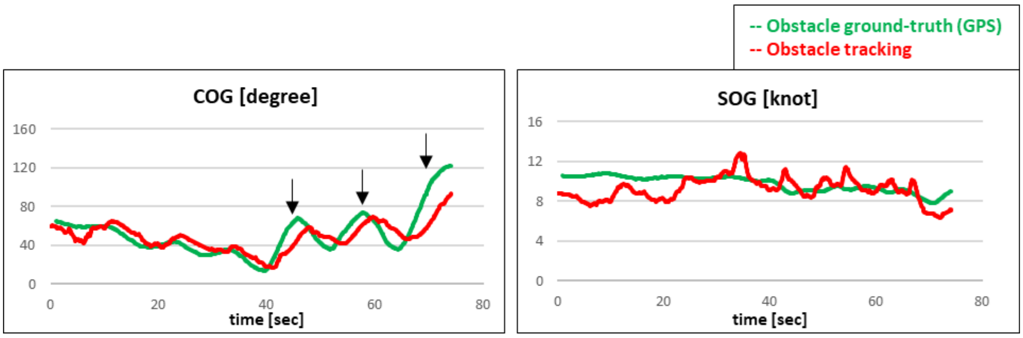


Figure 20. COG and SOG tracking result on 3-channel EO video

Second, the result of detecting an obstacle in 1-channel EO video in the same situation is shown in Figure 21. 1-channel EO video has a horizontal FOV of 63.0° and a resolution of 1280×720. Since the specifications of individual channels are the same as those of the previous 3-channel EO video, very similar detection results can be confirmed for this example.

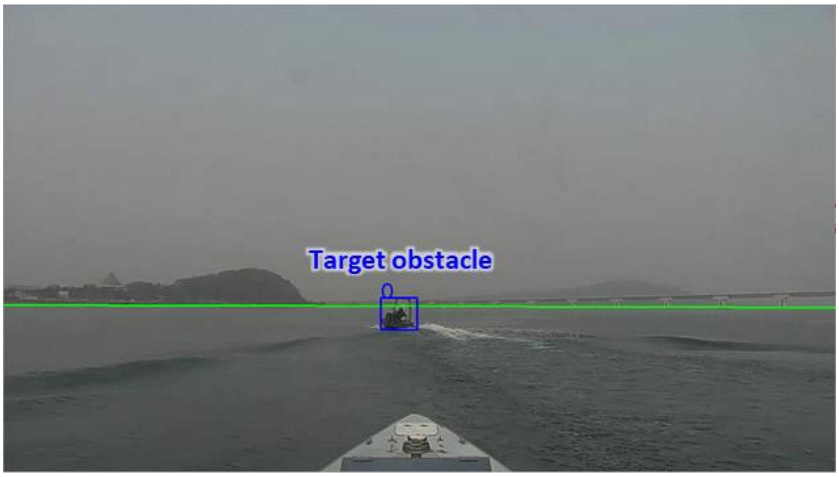


Figure 21. Detection result of target obstacle on 1-channel EO video

The result of tracking the trajectory of the obstacle in 1-channel EO video is shown in Figure 22. It shows very similar tracking characteristics to the previous 3-channel EO video, and the MAE was relatively large at 12.09m. Due to the posture maintenance built into the 1-channel EO camera, there was a difference in the posture between the camera and USV. As a result, the horizontal line on the image was not accurately detected, increasing the error. However, this also did not exceed 80% of the general GPS error, confirming that tracking was possible.

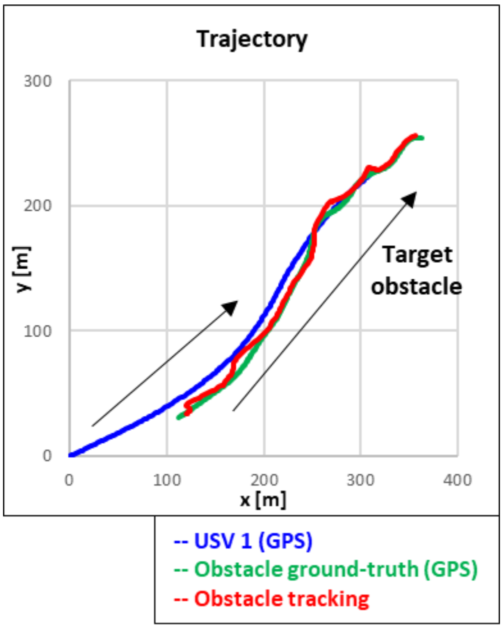


Figure 22. Trajectory tracking result on 1-channel EO video

The result of tracking the COG and SOG of the obstacle in 1-channel EO video is shown in Figure 23. The MAE of COG and SOG were 15.19° and 2.18 knots, slightly higher than the 3-channel EO video tracking results. It is also because of the same reason as the position error due to the self-posture maintenance described above. That is, when a horizon line is accurately detected without a separate posture-maintaining, it suggests that tracking error can be greatly reduced. Other tracking characteristics are similar to the previous 3-channel EO video, and the tracking delay due to the assumption of the constant velocity motion also appears similar.

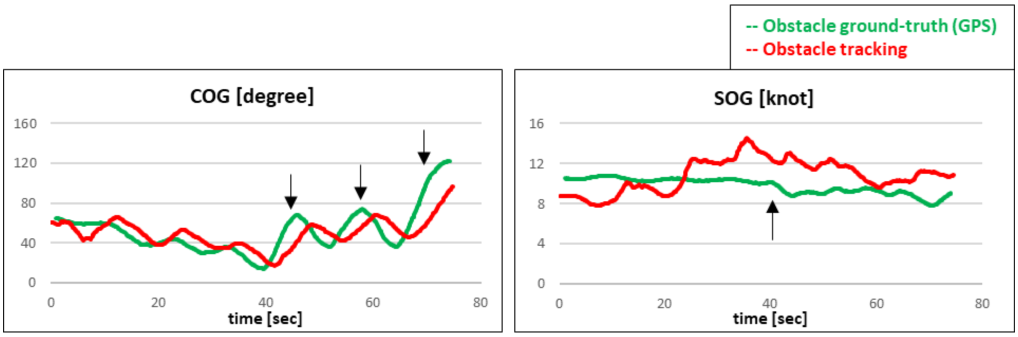


Figure 23. COG and SOG tracking result on 1-channel EO video

Finally, the result of detecting obstacles in 1-channel IR video is shown in Figure 24. For 1-channel IR video, the horizontal FOV is 35.5°, and the resolution is 720×480. It can be expected that the detection performance will be low due to the low image resolution compared to the EO camera. But the performance of detection algorithm was confirmed similarly because the horizontal FOV was inversely proportional to the decrease in resolution.

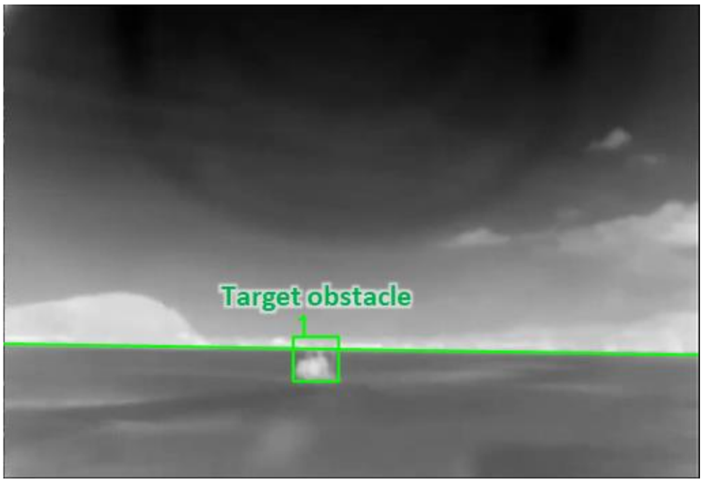


Figure 24. Detection result of target obstacle on 1-channel IR video

The result of tracking the trajectory of the obstacle in a 1-channel IR video is shown in Figure 25. Similar to the 1-channel EO camera, the IR camera has built-in posture maintenance, so there was the same problem the horizontal line was not accurately detected in some parts.

As a result of the tracking, the MAE was 9.22m, and the error was smaller than the tracking result of the 1-channel EO video where the same horizontal line detection error occurred. The reason for obtaining more accurate tracking results in poor-quality IR video was that the reduction in horizontal FOV was more dominant than the reduction in resolution. So more accurate detection within a small area in front of the USV was possible, and tracking accuracy was improved accordingly.

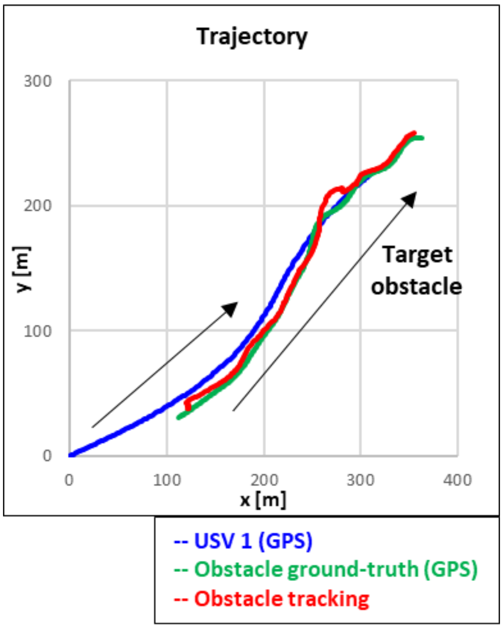


Figure 25. Trajectory tracking result on 1-channel IR video

The result of tracking the COG and SOG of the obstacle in a 1-channel IR video is shown in Figure 26. MAE was found to be COG 13.93° and SOG 1.83 knots, respectively. Although the horizontal line was not accurately detected for the same reason as the 1-channel EO video, the error is relatively small because the horizontal FOV is small.

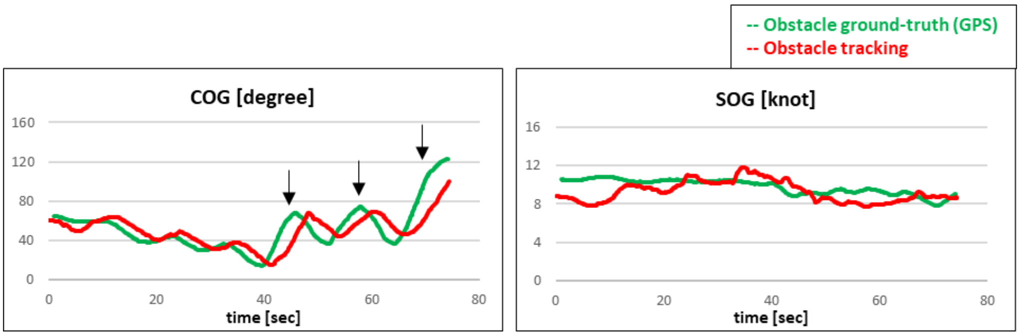


Figure 26. COG and SOG tracking result on 1-channel IR video

The tracking results from the images of three different types of cameras installed in the same location were fused using the sensor-to-sensor fusion method.

First, the trajectory of the obstacle is shown in Figure 27. As a result of fusion, the MAE was 8.43m, which is about 10.32% lower than the mean average error before fusion. Since the tracking results from all three cameras show almost similar tracking characteristics, there was no distinct difference. Partially, because a large position error does not occur compared to the tracking result before fusion, the fused trajectory of the obstacle appeared smooth.

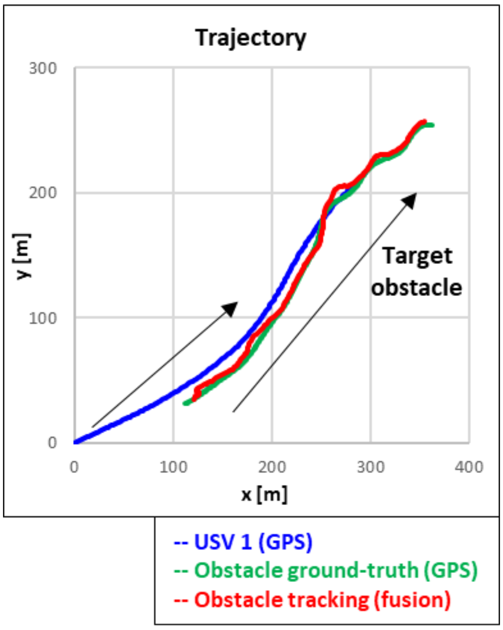


Figure 27. Trajectory fusion result

Next, the result of converging COG and SOG is shown in Figure 28. The MAE was COG 8.01° and SOG 1.85 knots, respectively, 10.50% and 6.09% lower than the average error before fusion. The convergence result of COG and SOG, like trajectory, did not show a big difference compared to the preceding camera-specific tracking result. Among the three types of camera tracking results, it showed the most similar aspect to the tracking result of the 3-channel EO camera, which had the highest tracking reliability. So through this fusion result, the convergence result that properly reflected the error covariance of the tracking result was derived.

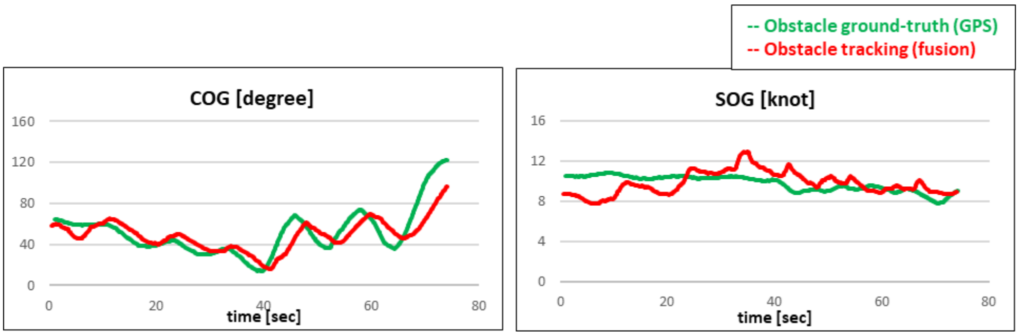


Figure 28. COG and SOG fusion result

In this case, obstacles were tracked from the images of three identically installed cameras, and the tracked data of the obstacle was fused. As a result, the error was reduced through the fusion, compared to the average tracking error of the three cameras. However, it was greater than the minimum error among the tracking errors of the three cameras. Since the tracking characteristics of the cameras are similar rather than opposite, it is difficult to improve the accuracy of the tracking result with the minimum error through fusion.

However, the camera sometimes has different tracking characteristics and errors. For example, EO cameras generate high errors in low-light environments and may not be able to track them at all. In contrast, IR cameras have the disadvantage that tracking is difficult in most areas due to a narrow horizontal FOV. Similarly, in the real world, the tracking accuracy of each camera may vary depending on the environment around the boat. So through a fusion technique that reflects the tracking reliability of different cameras, we can expect that tracking errors can be reached below a certain level without checking the tracking reliability of each camera every time.

5.2.2. Case 2: Collision avoidance

Case 2 is an example of detecting and tracking by defining the boat on the right side of Figure 15 as the target obstacle in a situation where three boats assume a collision. USV 1, the left boat in the figure, is equipped with the same 3-channel EO camera as Case 1, and USV 2, the boat approaching from the bottom in the figure, is equipped with the same 1-channel EO camera as Case 1. Since the collision situation is assumed, it takes little time to observe obstacles simultaneously in USV 1 and USV 2. However, it is a case in which fusion characteristics can be confirmed relatively well.

The result of detecting obstacles approaching head-on in USV 1 is shown in Figure 29. The obstacle detected in the left frame of the figure is the target obstacle, and the boat located in the middle frame is USV 2. Based on USV 1, the obstacle is approaching head-on from a position up to 100m away.

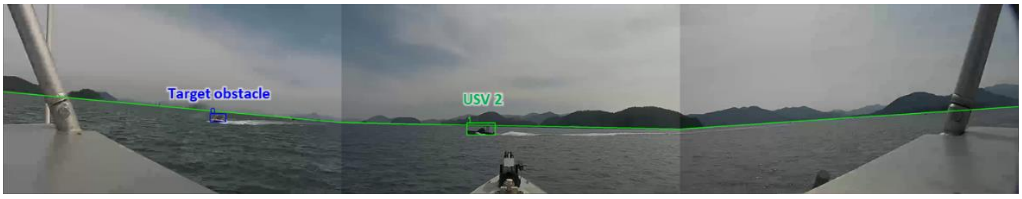


Figure 29. Detection result of target obstacle on USV 1

First, the trajectory that tracks the target obstacle from the viewpoint of USV 1 is shown in Figure 30. In the figure, it can be seen that the position error occurred in the direction parallel to the movement direction of the obstacle. That is, the trajectory of the obstacle oscillates forward and backward along the direction of movement.

The reason for such a tracking error is that the camera-based obstacle detection method proposed in this paper has a characteristic that the distance estimation error is relatively large compared to the bearing estimation error. As the obstacle gets farther away, the bearing error does not change significantly. In contrast, the distance error is affected by the decrease in the resolution of the obstacle in an image, so when the obstacle gets farther away, resulting in a large error. Therefore, from the viewpoint of USV 1 tracking in the movement direction of the obstacle, it can be seen that the distance error of the obstacle appears along the movement direction of the obstacle. In this case, MAE of the trajectory tracking result was 2.27m because of the short tracking time, resulting in a small error.

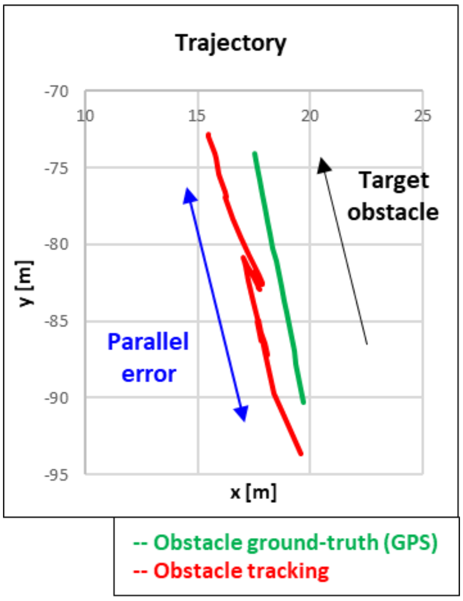


Figure 30. Trajectory tracking result on USV 1

The result of tracking the COG and SOG of the target obstacle in USV 1 is shown in Figure 31. As mentioned in the previous trajectory results, position of the obstacle showed an oscillation pattern along the movement direction of the obstacle. The SOG was also tracked as an oscillation pattern due to the effect. On the other hand, the direction of movement of the obstacle was tracked relatively accurately, and COG had little error. To sum it up, the obstacle was observed as moving in a constant direction but at oscillating speeds. MAE was 2.98° for COG and 0.64 knots for SOG.

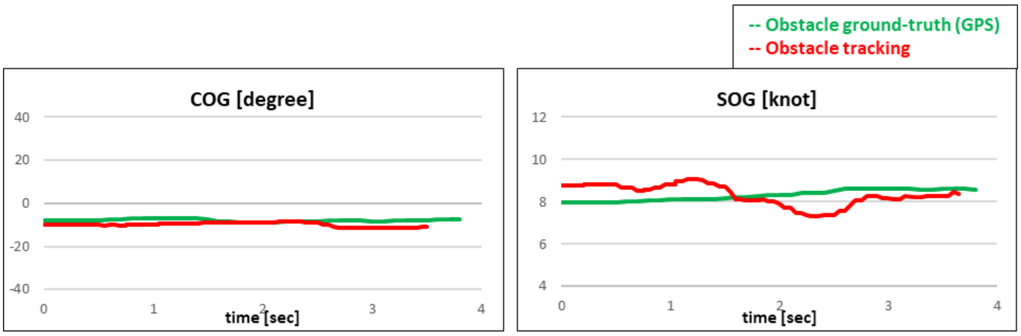


Figure 31. COG and SOG tracking results on USV 1

Figure 32 shows the result of detecting obstacles passing by crossing in USV 2. Although some water splashed on the camera, no false detection occurred in this example. To prevent false detection and a decrease in detection accuracy, we added additional training images (images with fog or water droplets) that impaired detection accuracy when training the obstacle detection algorithm. Based on USV 2, the obstacle starts from a position up to 80m away and passes through a crossing.

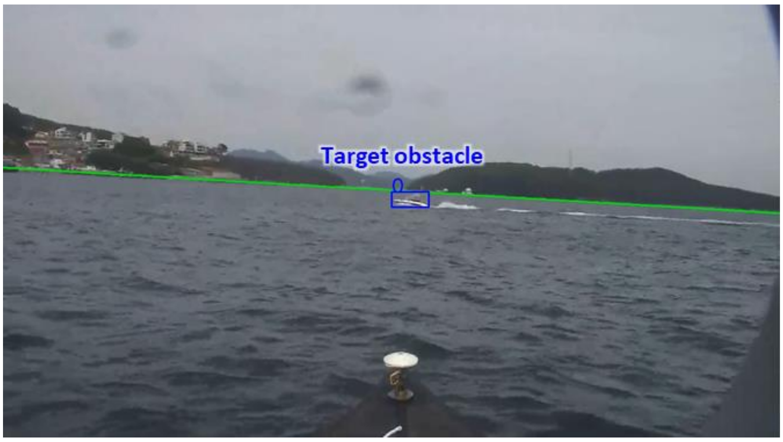


Figure 32. The detection result of target obstacle on USV 2

The result of tracking the trajectory of the obstacle in USV 2 is shown in Figure 33. It can be confirmed that it is different from the result of tracking the trajectory of the obstacle in USV 1 above. As a result of tracking from the viewpoint of USV 2, a large perpendicular way error occurred in the movement direction of the obstacle.

The characteristics of tracking error differ relative to the USV 1 because the direction error occurs on the positional relationship with the obstacle. In USV 1, a distance error occurred along the movement direction of the obstacle, resulting in a large position error and SOG error. In contrast, in USV 2, the distance error occurred in a direction perpendicular to the obstacle. Because USV 2 tracked the obstacle from the side of it. The tracking pattern is shown in the figure. The MAE of the tracking result was 2.89 m, almost similar to that of USV 1.

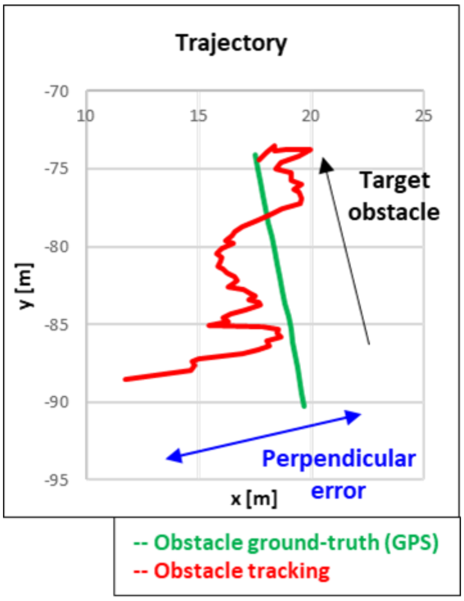


Figure 33. Trajectory tracking result on USV 2

The result of tracking the COG and SOG of the obstacle in USV 2 is shown in Figure 34. The tracking characteristic in USV 2 is the exact opposite of the one in USV 2. As an error occurred perpendicular to the movement direction of the obstacle, the COG was tracked as oscillating in all sections. As a result, the average error was larger than that of USV 1, where the COG error rarely occurred. In contrast, since the position error associated with the forward speed did not occur significantly, the SOG error was relatively low than that of USV 1. MAE was 6.78° for COG and 0.42 knots for SOG.

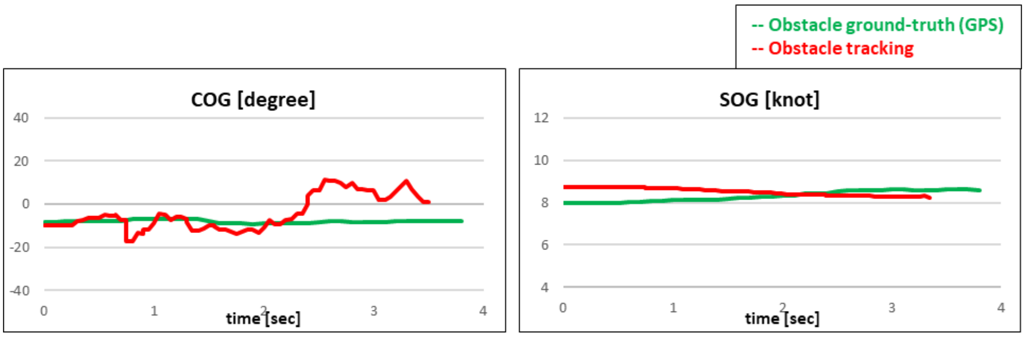


Figure 34. COG and SOG tracking results on USV 2

The result of tracking the same obstacle from different viewpoints (front/side) was fused in real time through the sensor-to-sensor fusion method. First, the result of fusion with each other is shown in Figure 35. Compared to the individual tracking results, it can be seen that both parallel errors and perpendicular errors are reduced. In other words, when two tracking results with different tracking characteristics due to the difference in viewpoint are fused, the disadvantages found in each tracking result can be compensated. Numerically, the MAE after fusion was 1.78m, a decrease of 31.0% compared to before fusion.

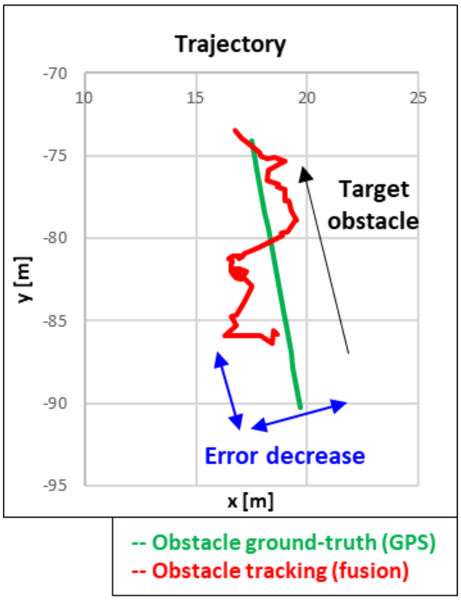


Figure 35. Trajectory fusion result

Figure 36 shows the COG and SOG of obstacles estimated through fusion. Compared to the previous individual tracking results, the error is significantly reduced. First, in the case of COG, a large error occurred in USV 2 and a relatively small error in USV 1. However, after fusion, the error from USV 2 is decreased by the influence of USV 1, which has high tracking reliability. Also, in the case of SOG, some tracking error tendency of USV 1 remains, but the oscillation pattern disappeared after fusion. As a result, MAE decreased by about 38.9% compared to before fusion to COG 2.98°, and SOG was 0.57 knots.

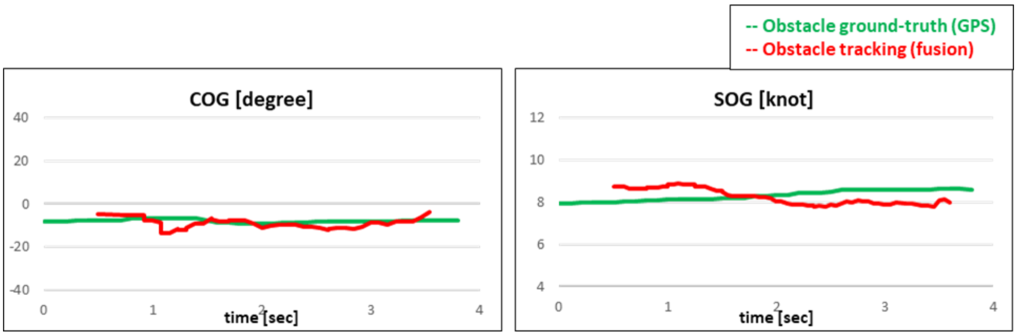


Figure 36. COG and SOG fusion result

In this case, the results of tracking the same obstacle from different viewpoints were fused to obtain more accurate motion data of the obstacle. The disadvantages appeared in tracking data from different viewpoints compensated each other through fusion. Parallel errors and perpendicular errors were large in each viewpoint, respectively, based on the direction of the obstacle, but both directions of errors decreased after fusion.

In the real situation, if communication between operating USVs is possible, more accurate maritime obstacle recognition will be possible by fusing obstacle data tracked at various viewpoints in time by using the proposed fusion method.

5.2.3. Case 3: Platooning

In the last case, four boats operate in a platoon. As shown in Figure 15, the tracking was performed by defining the boat at the rear is called USV 1, the boat at the forefront as USV 3, and the boats on the left and right are target obstacle 1 and target obstacle 2, respectively. In USV 1, a 3-channel EO camera was installed to look forward, and in USV 3, two 3-channel EO cameras capable of monitoring in whole directions were installed. The camera mounted on USV 1 has the same specifications as the 3-channel EO camera of the previous case. The two cameras on USV 3 each have a horizontal FOV of 180°, and the image resolution is 2160 × 480. Using two cameras that can monitor 180°, it is possible to detect and track target obstacle 1 and target obstacle 2 located in the rear.

First, the result of detecting target obstacle 1 and target obstacle 2 in USV 1 is shown in Figure 36. The boat surrounded by the blue detection box in the left frame is target obstacle 1, and the boat surrounded by the green detection box in the right frame is target obstacle 2. The tracking and fusion results were analyzed only for the part where simultaneous detection and tracking were possible because they existed within the field of view of USV 1 and 3 during platoon operation.

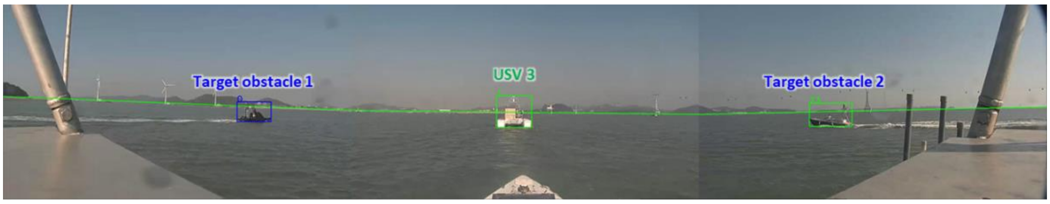


Figure 37. Detection result of target obstacles on USV 1

The result of tracking the trajectory of target obstacle 1 in USV 1 is shown in Figure 37. The platoon operation was conducted for a relatively long time of more than 3 minutes, but it was the result of tracking until the target obstacle 1 turned to the left and left the field of view of USV 1.

Although there was almost no position error in the straight-line section, a position error occurred when target obstacle 1 made a sharp turn to the left because it could not properly track the sharp turn. Since the constant velocity motion was assumed in the tracking filter, the acceleration due to the turn could not be properly reflected, resulting in a relatively large position error. However, compared to the previous cases, the MAE was relatively small at 4.51m because it was tracking at a distance of about 60m, which is a relatively short distance.

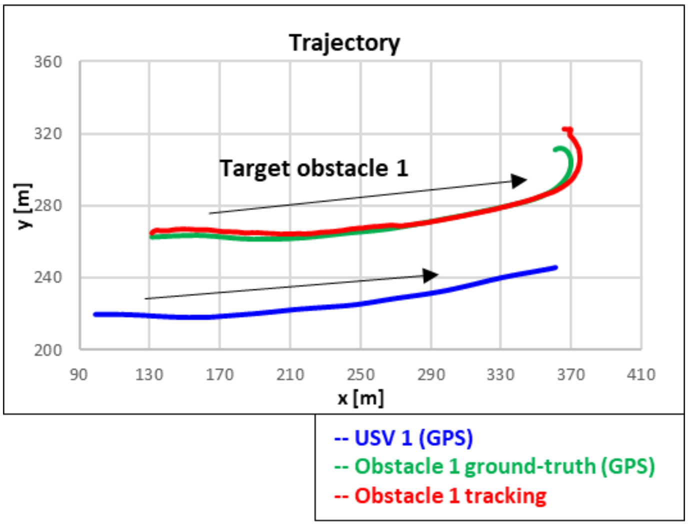


Figure 38. Trajectory tracking result of target obstacle 1 on USV 1

The result of tracking the COG and SOG of target obstacle 1 in USV 1 is shown in Figure 38. As can be seen in the figure, similar to the trajectory tracking results, the COG and SOG were well tracked in the section where target obstacle 1 moved in a straight line. However, in the section where the obstacle turning occurred after 55 seconds, the sudden turning acceleration was not sufficiently reflected, so some delay appeared in the COG and SOG tracking results. The calculated MAE seemed that the proposed tracking algorithm tracked accurately with 13.25° and 0.94 knots, respectively.

However, the maximum error was 101.04° and 4.05 knots, which caused a very large error temporarily due to inaccurate tracking in the turning section. Through these results, the turning motion of a small boat can temporarily generate a very large tracking error. Therefore, we must consider introducing a tracking filter assuming a constant acceleration motion to reduce the error and delay.

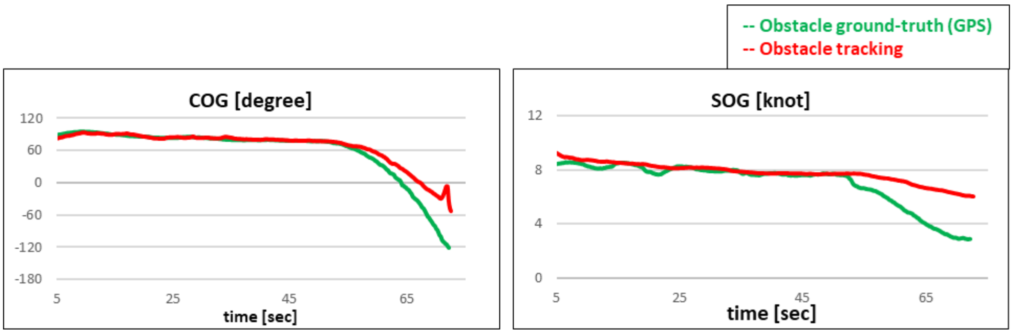


Figure 39. COG and SOG tracking result of target obstacle 1 on USV 1

Figure 40 results from tracking the trajectory of target obstacle 2 in USV 1. As seen in the figure, same as the tracking result of target obstacle 1 above, some delay occurred in the section where target obstacle 2 makes a sharp turn to the right. It occurred for the same reason as explained in the tracking result of target obstacle 1. However, in the case of target obstacle 2, the turning path was relatively gentle, so the delay was relatively small. In addition, some errors occurred in the straight motion section. It occurred because a part of target obstacle 2 was occluded by the structure of USV 1 existing in the right frame of Figure 37. As a result, a positional error occurred in the straight motion section. The MAE was 4.04m, similar to target obstacle 1.

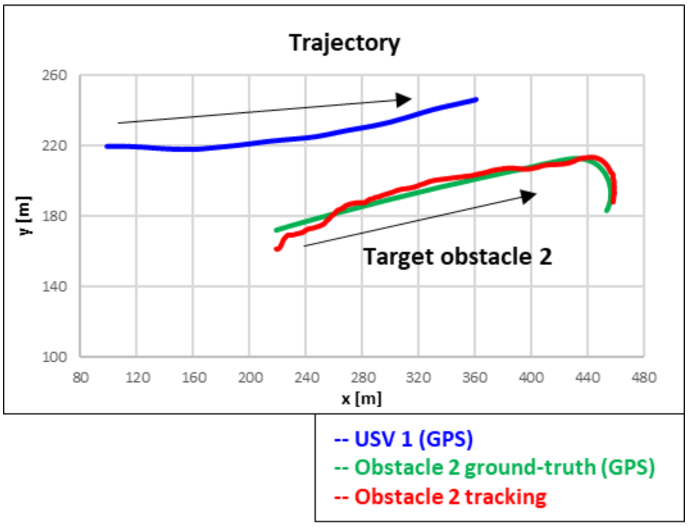


Figure 40. Trajectory tracking result of target obstacle 2 on USV 1

The result of tracking the COG and SOG of target obstacle 2 in USV 1 is shown in Figure 40. First, looking at the COG graph, there was an aspect in which some tracking results vibrated in the straight motion section due to occlusion by the USV 1 structure, and a delay occurred in the tracking result when the obstacle turned to the right. However, compared to the previous tracking result of target obstacle 1, it can be seen that the delay occurred less because the turning radius was smaller.

Next, SOG could not track all the detailed speed increases and decreases of target obstacle 2, but the overall trend was tracked with reasonable accuracy. The MAEs were 11.26° and 1.32 knots, respectively.

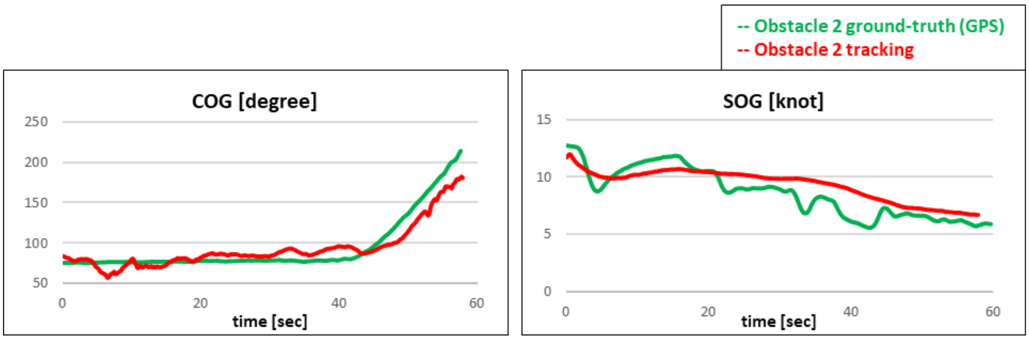


Figure 41. COG and SOG tracking result of target obstacle 2 on USV 1

Next, the result of tracking the target obstacles from the viewpoint of USV 3 in the same case is shown in Figure 41. The three frames on the left half are the areas corresponding to the front 180°, and the three on the right half are the areas corresponding to the rear 180°. Among the obstacles detected from the rear of USV 3, the rightmost blue detection box is target obstacle 1, second green detection box located from the right is USV 1, and the third pink detection box located from the right is target obstacle 2. It is similar to USV 1 above, but it can be seen that the camera installation height of USV 3 is relatively high at 6.0m, and the obstacle is detected as relatively small.

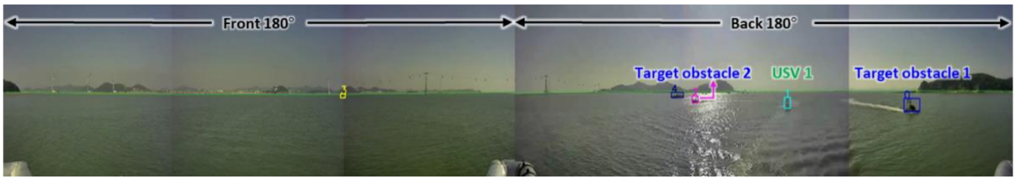


Figure 42. The detection result of target obstacles on USV 3

The result of tracking the trajectory of target obstacle 1 in USV 3 is shown in Figure 43. The tracking time is slightly longer than in USV 1 above. Shifting occurred during the whole tracked path compared to ground truth, but the magnitude was not large. As in USV 1, the target obstacle 1 was not accurately tracked when the obstacle turning motion was sharp, resulting in a large error. It can be seen that the MAE is 8.62 m, which is relatively large compared to the previous tracking results in USV 1.

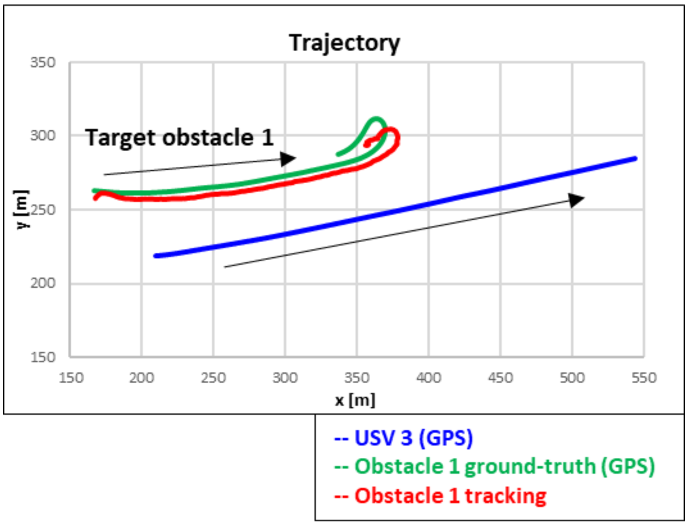


Figure 43. Trajectory tracking result of target obstacle 1 on USV 3

The result of tracking the COG and SOG of target obstacle 1 in USV 3 is shown in Figure 44. Shifting occurred in trajectory tracking, but other tracking aspects except shift were similar to the ground truth, so COG and SOG tracking results were also tracked similarly to ground truth. As in USV 1, tracking was performed with very small errors in both COG and SOG in the straight motion section, and some delay appeared in the sharp turn section. USV 3 has a slightly longer tracking length than USV 1, so we can better see the delay.

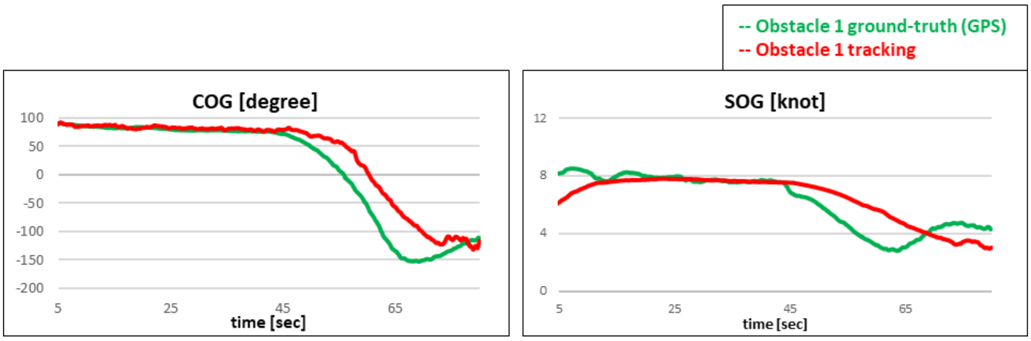


Figure 44. COG and SOG tracking results of target obstacle 1 on USV 3

The result of tracking the trajectory of target obstacle 2 in USV 3 is shown in Figure 45. Similar to the trajectory of target obstacle 1, shifting occurred in the entire path, but the overall tendency of the tracked trajectory followed the ground truth well. In addition, the turning radius of target obstacle 2 is relatively large compared to that of target obstacle 1, so the tracking error caused by turning acceleration is smaller than that of target obstacle 1 in the sharp turning section. The MAE was relatively large at 10.86 m due to shifting. However, more accurate tracking will likely be possible if shifting is removed by improving horizon line detection.

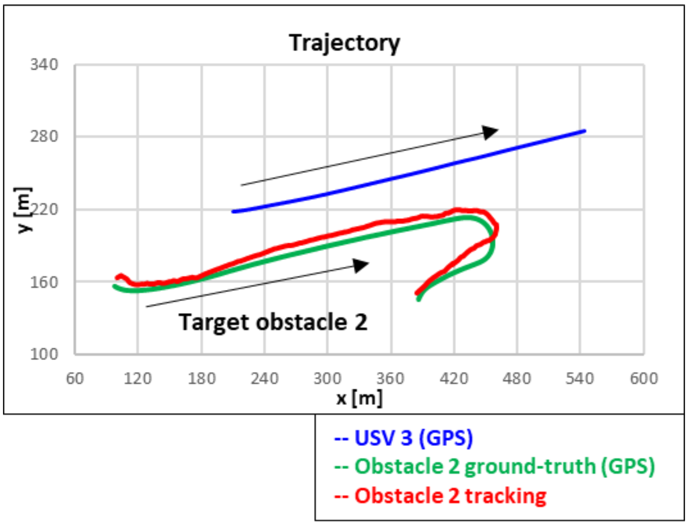


Figure 45. Trajectory tracking result of target obstacle 2 on USV 3

The result of tracking the COG and SOG of target obstacle 2 in USV 3 is shown in Figure 46. Tracking is performed well in whole tracking sections. Even in the section where target obstacle 2 made a sharp turn, the delay in tracking the COG of the obstacle occurred less compared to the tracking result of target obstacle 1. Moreover, in the case of SOG, no delay was observed. The MAEs were 10.59° and 1.01 knots, respectively, similar to those tracked in USV 1.

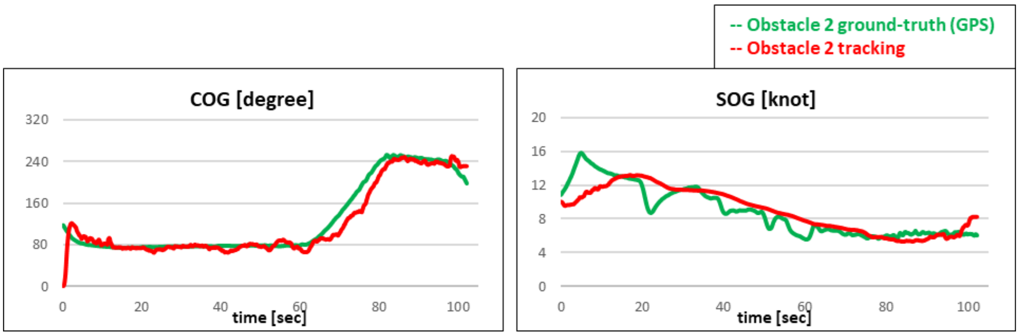


Figure 46. COG and SOG tracking results of target obstacle 2 on USV 3

Figure 47 shows the trajectory that fuses the tracking results from the front and rear viewpoints of target obstacle 1. The fusion was conducted with the sensor-to-sensor fusion method. Compared to Figure 38 and Figure 43, the tracked trajectory seems more similar to the ground truth, and the tracking error decreased even in the sharp turn section, where the tracking error was large in tracked individuals.

Although the error in the sharp turn section occurred largely in the previous two tracking results, the error decreased after a fusion because the errors generated in the two tracking results had opposite characteristics. In detail, when tracking in USV 1, shifting occurred in the direction away from the USV, and an error occurred. Whereas, when tracking in USV 3, shifting occurred in a direction closer to the USV, and an error occurred. It shows that when tracking results with opposite error characteristics are fused, the errors are compensated, and more accurate tracking results can be derived. The MAE was 4.42m, a 32.7% decrease compared to before fusion.

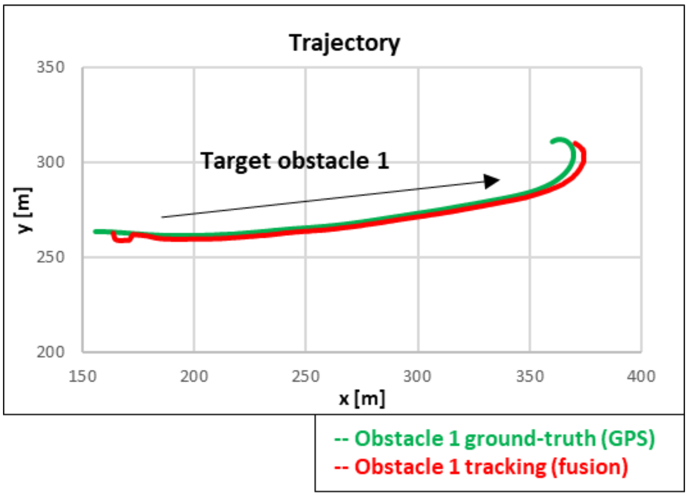


Figure 47. Trajectory fusion result of target obstacle 1

The result of fusing the COG and SOG of target obstacle 1 is shown in Figure 48. The tracking results from USV 1 and USV 3 were almost similar, so the fusion results of COG and SOG were not significantly different. Unlike trajectory, tracking characteristics opposite to each other did not appear, so a fusion effect beyond the appropriate fusing of two similar tracking results did not appear. MAEs were 11.31° and 0.70 knots, respectively. This fusion result shows that fusion between tracking results that do not have mutually opposite characteristics or compensating characteristics has a limitation in not improving the tracking result.

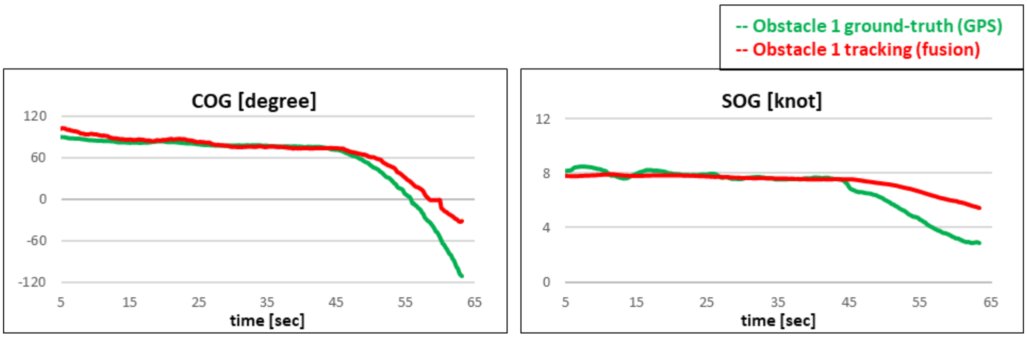


Figure 48. COG and SOG fusion results of target obstacle 1

Figure 49 shows the trajectory that fused the tracking results from the front and rear of target obstacle 2, respectively. Similar to the fusion result for target obstacle 1, a relatively smooth tracking result was generated compared to before fusion. However, some shifting occurring due to the effect of the tracking result in USV 3 still needs to be eliminated. The MAE was 6.03m, a decrease of 19.1% compared to before fusion, but greater than that of USV 1.

In this case, an accurate tracking result may be contaminated if the two tracking results do not have opposite characteristics or errors are concentrated in one of the two tracking results. Therefore, it is necessary to study fusion algorithms by carefully analyzing the correlation of tracking results using different cameras to prevent contamination of accurate tracking results through fusion. For example, in this case, if USV 1 has an accurate tracking result but tends to have a large standard deviation of the distance, and USV 3 has a small standard deviation of the distance but tends to shift the average. More accurate fusion results are expected to be obtained through the fusion algorithm that reflects the correlation.

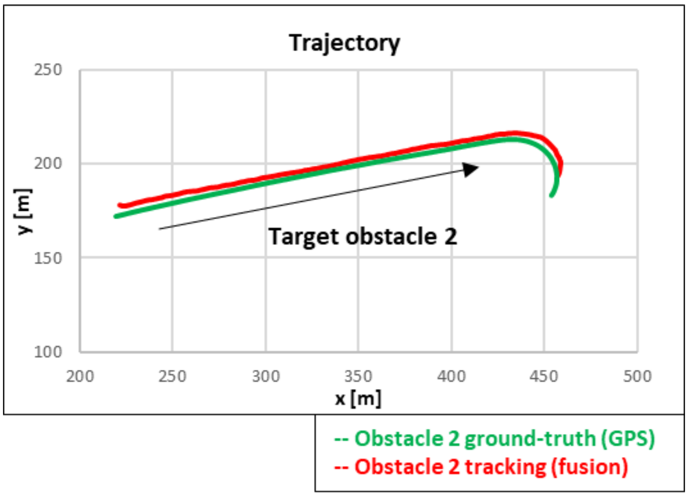


Figure 49. Trajectory fusion result of target obstacle 2

The result of fusing the COG and SOG of target obstacle 2 is shown in Figure 50. Similar to the fusion results of target obstacle 1, the results of COG and SOG were not significantly different from those before fusion. In the case of COG, it was accurately tracked in the straight motion section, but a delay occurred in the sharp turn section, and in the case of SOG, it followed the overall trend well. The MAEs were 11.28° and 0.99 knots, respectively, similar to the values before fusion.

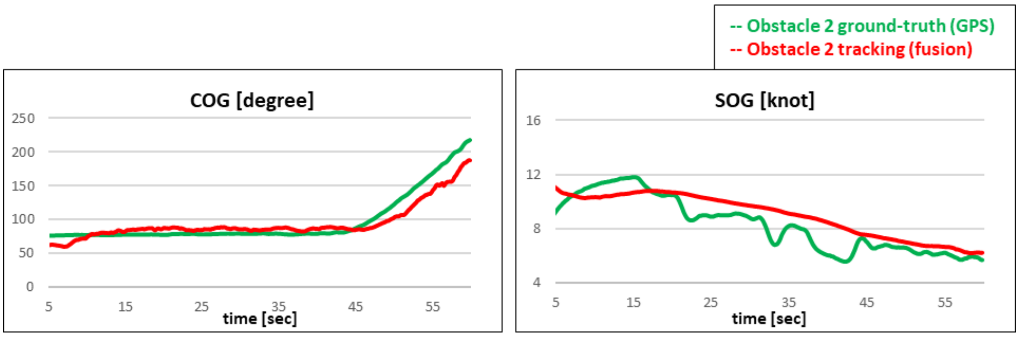


Figure 50. COG and SOG fusion results of target obstacle 2

In this case, the results of tracking obstacles from different viewpoints were fused. In the case of opposite error characteristics between different tracking results, the error was greatly reduced through fusion, and tracking results close to the ground truth could be obtained. However, when a large error occurred in one of the different tracking results, the tracking result with the minimum error could be contaminated due to fusion with inaccurate tracking results. Through this, there is a need for a fusion method that can reflect the error characteristics of each tracking result by closely analyzing the correlation between the tracking results through different cameras or sensors.

5.2.4. Summary

The camera-based recognition technology proposed in this paper was applied to three cases, and the results are summarized in Table 4 below. In all cases, the tracking results showed convergence within about 18 m with an accuracy of 80% of the general GPS error. And in most cases, the accuracy of tracking data was reduced after the fusion of tracked data from different camera conditions. Through the application, it can be proved that the camera image-based tracking is reliable, and the fusion was useful in situations where communication between the USVs is possible.

First, in case 1, obstacles were detected and tracked by three types of cameras installed on the boat, and each tracking data was fused with the other. Tracked data from three different cameras did not show significant differences due to clear weather. Through this, stable tracking could be possible through fusion without a separate reliability evaluation process when various cameras with different operating environments are simultaneously operated regardless of daytime, nighttime, or weather.

Case 2 detected and tracked an obstacle moving in a straight line from the front and side, and simultaneously tracked data was fused. Through this, different tracking characteristics, because of the relatively large distance error compared to the bearing error, could appear depending on the observation point of the obstacle. In addition, when tracking data is fused in this situation, the characteristics that are disadvantages of each tracking data can be compensated with each other according to the reliability of tracking data.

Finally, in case 3, obstacles were detected and tracked from the front and rear in a platooning situation, and the tracking data was fused. Through this, when opposite error characteristics between different tracking data appear, the error can be greatly reduced after fusion. On the other hand, when a relatively large error occurs in one tracking data, tracking data with a relatively small error is contaminated. That is, it is necessary to analyze each tracking characteristics of the camera or the target object in more detail and develop a fusion algorithm that reflects them.

Table 4. Summary of application case result

Application case		Camera	Trajectory MAE [m]	COG MAE [°]	SOG MAE [knots]
Case 1		3-channel EO	6.58	12.45	1.40
		1-channel EO	12.09	15.19	2.18
		1-channel IR	9.22	13.93	1.83
		Fusion	**8.43**	**8.01**	**1.85**
Case 2		USV 1	2.27	2.98	0.64
		USV 2	2.89	6.78	0.42
		Fusion	**1.78**	**2.98**	**0.57**
Case 3	Obstacle 1	USV 1	4.51	13.25	0.94
		USV 3	8.62	17.57	0.88
		Fusion	**4.42**	**11.31**	**0.70**
	Obstacle 2	USV 1	4.04	11.26	1.32
		USV 3	10.86	10.59	1.01
		Fusion	**6.03**	**11.28**	**0.99**

6. Conclusions and future works

6.1. Conclusions

In this paper, we implemented a maritime obstacle detection algorithm based on YOLOv5, a deep learning algorithm capable of real-time maritime obstacle detection for actual USV applications. In addition, CBAM was introduced to improve the accuracy of the YOLOv5 algorithm. As a result of evaluating the AP accuracy for actual marine images, the EO detection algorithm improved from 95.75% to 95.98%, and the IR detection algorithm showed a large improvement from 94.44% to 95.67%. Both algorithms proved that the obstacle detection accuracy was over 90%, which is not enough to perform detection-based tracking.

A position conversion method using a monocular camera was proposed to perform obstacle tracking based on the detected obstacle data. First, the position vector of the detected obstacle was defined. Then using a defined position vector, the bearing of the obstacle was estimated through quaternion rotation transformation. Finally, the distance between the obstacle and the boat was estimated by detecting a horizontal line on the image. Calculated bearing and distance is a value representing the relative position of an obstacle defined by the own USV and was used as a sensor measurement in the obstacle tracking process.

To estimate the motion data of an obstacle, an adaptive extended Kalman filter suitable for camera-based obstacle detection results is proposed. In the case of detecting an obstacle based on the camera image, the error variability was small for objects near. However, the variability was high for objects at a long distance. Therefore, to consider these characteristics in the tracking phase, the adaptive error covariance was defined by analyzing the obstacle detection results. To define the adaptive error covariance, the bounding box variability of the obstacle detection algorithm was analyzed, and the resulting variability of the actual distance estimation result was analyzed.

As a result of verifying the obstacle tracking method proposed in this paper based on images and GPS data acquired from the actual sea, it was confirmed that it converged within 80% of the GPS error for all application cases. In addition, the COG and SOG tracking results showed that the overall movement tendency was tracked well, except for the delay that occurred when the obstacle made a sharp turn.

To obtain more accurate tracking data based on the data tracked by each camera, data tracked by multiple cameras and multiple boats were fused. Fusion was largely applied and verified in two cases: a case in which data tracked by three types of cameras mounted on a boat is fused, and a case in which data tracked by two boats observing obstacles from different viewpoints is fused.

First, it was confirmed that more stable tracking data could be obtained through fusion between cameras with different operating environments. However, drastic error reduction did not appear when the fusion of data tracked with similar accuracy and characteristics was performed. In addition, it was confirmed that when the data tracked by the two boats at different viewpoints are fused, the distance estimation errors that appear differently depending on the viewpoints compensate each other, and the errors can be largely reduced.

However, on the contrary, if the error is weighted on only one of the two tracking data to be fused, the accurate tracking data can be contaminated through fusion with the less accurate data. So, in this case, we confirmed that there is a need to analyze the relationship between tracking data between different cameras.

6.2. Future works

To improve the multi-image-based maritime obstacle detection and tracking method proposed in this paper through future research, tracking errors due to occlusion that may occur in the detection stage must first be improved. As shown in Figure 19, occlusion due to the wake may occur, but in the actual sea, errors due to occlusion between obstacles occur more frequently, as shown in Figure 51. Therefore, such detection errors must be reduced first in the obstacle detection algorithm. Furthermore, even if detection errors occur, tracking management, such as backing up tracking data during occlusion, as shown in Figure 52, has to be devised to avoid incorrect detection results contaminating tracking data.

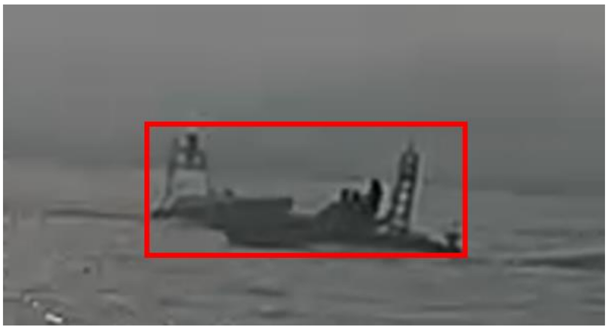


Figure 51. False detection result by occlusion

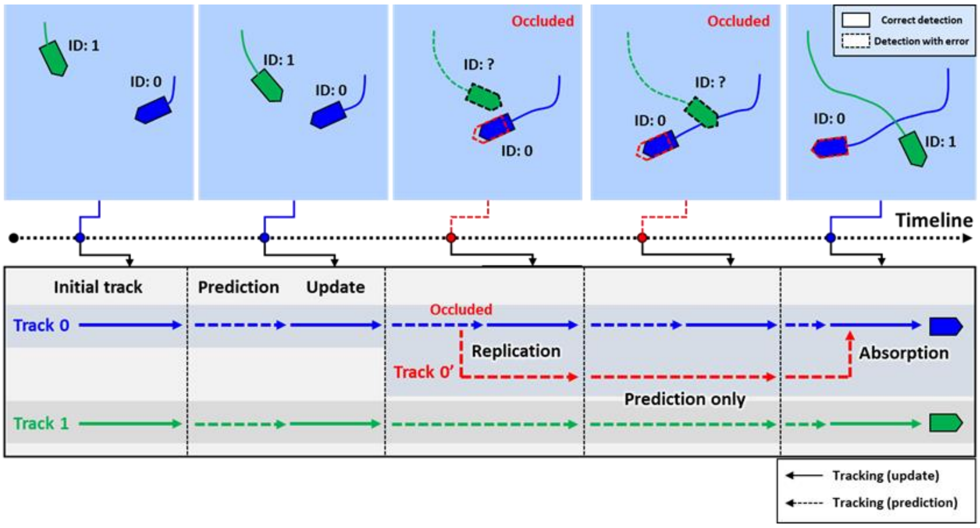


Figure 52. Example of track management when occlusion happened

Also, as mentioned many times, it is necessary to carefully analyze the correlation between data tracked by different cameras. The sensor fusion algorithm used in this paper, fast CI, is designed to produce optimal fusion results between two tracking data with ambiguous correlation. However, as a result of applying this to this paper, it was found that when a relatively large error occurs in only one of the fusion data, it is not properly reflected. In other words, if the reliability of one side has significantly decreased through correlation analysis between cameras, it is necessary to improve the algorithm, such as excluding it from the fusion target or giving weight to the reliability of the side with less error.

References

[1] L. Zhu, X. Geng, Z. Li, C. Liu, Improving yolov5 with attention mechanism for detecting boulders from planetary images, Remote Sens. 13 (2021). https://doi.org/10.3390/rs13183776.

[2] GitHub, YOLOV5-Master, (2021). https://doi.org/https://doi.org/10.5281/zenodo.4679653.

[3] S. Woo, J. Park, J.Y. Lee, I.S. Kweon, CBAM: Convolutional block attention module, in: Lect. Notes Comput. Sci. (Including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics), 2018. https://doi.org/10.1007/978-3-030-01234-2_1.

[4] Q. Wang, B. Wu, P. Zhu, P. Li, W. Zuo, Q. Hu, ECA-Net: Efficient channel attention for deep convolutional neural networks, in: Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit., 2020. https://doi.org/10.1109/CVPR42600.2020.01155.

[5] C. Fu, R. Duan, E. Kayacan, Visual tracking with online structural similarity-based weighted multiple instance learning, Inf. Sci. (Ny). 481 (2019). https://doi.org/10.1016/j.ins.2018.12.080.

[6] F. Rezaeianaran, R. Shetty, R. Aljundi, D.O. Reino, S. Zhang, B. Schiele, Seeking Similarities over Differences: Similarity-based Domain Alignment for Adaptive Object Detection, in: Proc. IEEE Int. Conf. Comput. Vis., 2021. https://doi.org/10.1109/ICCV48922.2021.00907.

[7] W. Zhang, X. zhong Gao, C. fu Yang, F. Jiang, Z. yuan Chen, A object detection and tracking method for security in intelligence of unmanned surface vehicles, J. Ambient Intell. Humaniz. Comput. 13 (2022). https://doi.org/10.1007/s12652-020-02573-z.

[8] J. Redmon, A. Farhadi, YOLOv3: An Incremental Improvement, (2018). http://arxiv.org/abs/1804.02767.

[9] J. Han, Y. Cho, J. Kim, J. Kim, N. Son, S.Y. Kim, Autonomous collision detection and avoidance for ARAGON USV: Development and field tests, J. F. Robot. 37 (2020) 987–1002. https://doi.org/10.1002/rob.21935.

[10] W.J. Lee, M. Il Roh, H.W. Lee, J. Ha, Y.M. Cho, S.J. Lee, N.S. Son, Detection and tracking for the awareness of surroundings of a boat based on deep learning, J. Comput. Des. Eng. 8 (2021). https://doi.org/10.1093/jcde/qwab053.

[11] A. Bewley, Z. Ge, L. Ott, F. Ramos, B. Upcroft, Simple online and realtime tracking, in: Proc. - Int. Conf. Image Process. ICIP, 2016. https://doi.org/10.1109/ICIP.2016.7533003.

[12] T. Kim, T.H. Park, Extended kalman filter (Ekf) design for vehicle position tracking using reliability function of radar and lidar, Sensors (Switzerland). 20 (2020). https://doi.org/10.3390/s20154126.

[13] D. Fränken, A. Hüpper, Improved fast covariance intersection for distributed data fusion, in: 2005 7th Int. Conf. Inf. Fusion, FUSION, 2005. https://doi.org/10.1109/ICIF.2005.1591849.

[14] H.B. Mitchell, Multi-Sensor Data Fusion: An Introduction, Springer International Publishing, 2007.

국문 초록

다중 영상 기반 해상 장애물 탐지 및 추적 방법

해양에서 발생하는 사고의 원인 중에서 사람의 과실이 비교적 높은 비율을 차지하고 있으며, 그에 따라 선박 주변 인지를 위한 자율 인지 시스템의 필요성이 대두되고 있다. 전통적인 인지 센서인 Automatic Identification System (AIS)와 Radio Detection and Ranging (RADAR)를 활용한 자율 인지 기술에 관한 연구가 활발히 이루어지고 있으나, USV 와 같은 소형선이 활동하는 연안에서는 AIS 누락, RADAR 사각 지대 등의 한계가 존재한다. 따라서 이들을 대체할 수 있는 새로운 인지 기술을 개발하고자 하였다.

본 논문에서는 전통적인 인지 센서의 한계를 보완하고 인간의 시각을 대체하고자 카메라를 활용한 자율 인지 기술을 제안하였다. 먼저 카메라 영상 기반의 실시간 객체 탐지 모델인 YOLOv5 모델을 개선하여 장애물 탐지 정확도를 높였으며, 단안 카메라를 활용한 위치 변환 알고리즘으로 탐지된 장애물의 상대적 위치를 추정하였다. 추정된 장애물의 상대적 위치를 바탕으로 본 논문에서 제안한 적응형 확장 칼만 필터 (adaptive extended Kalman Filter)를 이용하여 장애물의 운동 정보인 trajectory, Course Over Ground (COG), 그리고 Speed Over Ground (SOG)를 추정하였다. 또한 USV 가 전략적인 목적으로 운용되어 상호 간의 통신 및 협력이 가능할 경우를 가정하여 추적 정보의 정확도를 높이기 위해 서로 다른 카메라에서 추적된 정보 간의 센서 융합 (sensor fusion)을 수행하였다. 여러 대의 카메라 각각에서 추적된 정보를 서로 융합하여 추적 정확도를 개선하거나 개별 카메라의 추적 과정에서 발생하는 단점을 서로 상쇄하는 등 보다 정확한 추적 정보를 얻을 수 있음을 확인하였다.

주요어: Object detection (장애물 탐지), Object tracking (장애물 추적), Adaptive tracking filter (적응형 추적 필터), Data association (데이터 연관), Sensor fusion (센서 퓨전)

학번: 2021-21642