

Article

Real-Time Ship Tracking under Challenges of Scale Variation and Different Visibility Weather Conditions

Hu Liu ¹, Xueqian Xu ², Xinqiang Chen ^{3,*} , Chaofeng Li ³  and Meilin Wang ³

¹ School of Naval Architecture and Maritime, Zhejiang Ocean University, Zhoushan 316022, China; liuhu@zjou.edu.cn

² Intelligent Transport Systems Research Centre (ITSC), Wuhan University of Technology, Wuhan 430063, China; xuxueqian@stu.shmtu.edu.cn

³ Institute of Logistics Science and Engineering, Shanghai Maritime University, Shanghai 201306, China; cfli@shmtu.edu.cn (C.L.); wangmeilin@stu.shmtu.edu.cn (M.W.)

* Correspondence: chenxinqiang@stu.shmtu.edu.cn

Abstract: Visual ship tracking provides crucial kinematic traffic information to maritime traffic participants, which helps to accurately predict ship traveling behaviors in the near future. Traditional ship tracking models obtain a satisfactory performance by exploiting distinct features from maritime images, which may fail when the ship scale varies in image sequences. Moreover, previous frameworks have not paid much attention to weather condition interferences (e.g., visibility). To address this challenge, we propose a scale-adaptive ship tracking framework with the help of a kernelized correlation filter (KCF) and a log-polar transformation operation. First, the proposed ship tracker employs a conventional KCF model to obtain the raw ship position in the current maritime image. Second, both the previous step output and ship training sample are transformed into a log-polar coordinate system, which are further processed with the correlation filter to determine ship scale factor and to suppress the negative influence of the weather conditions. We verify the proposed ship tracker performance on three typical maritime scenarios under typical navigational weather conditions (i.e., sunny, fog). The findings of the study can help traffic participants efficiently obtain maritime situation awareness information from maritime videos, in real time, under different visibility weather conditions.

Keywords: visual ship tracking; scale-adaptive kernelized correlation filter; poor visibility condition; maritime situation awareness; smart ship



Citation: Liu, H.; Xu, X.; Chen, X.; Li, C.; Wang, M. Real-Time Ship Tracking under Challenges of Scale Variation and Different Visibility Weather Conditions. *J. Mar. Sci. Eng.* **2022**, *10*, 444. <https://doi.org/10.3390/jmse10030444>

Academic Editor: Mihalis Golias

Received: 27 February 2022

Accepted: 15 March 2022

Published: 20 March 2022

Publisher's Note: MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.



Copyright: © 2022 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

Smart ships are considered as fundamental facilities in the green shipping era, offering the potential of significantly reducing carbon emissions, loading larger cargos, automatically making navigation decisions, etc. [1–3]. Much research attention has been given to collecting on-spot traffic data from varied maritime sensors, which are further processed to fulfill maritime situation awareness tasks. Previous studies suggest that spatial-temporal ship positions provide crucial ship kinematic information, which provides elementary maritime information support for traffic situation awareness [4–6]. The ship position dataset can be easily obtained due to a large number of sensors (e.g., maritime surveillance cameras and automatic identification system (AIS) facilities) that are widely deployed both onboard and on off-shore buildings, and thus ship position streaming data is continuously obtained [7]. The long-range identification and tracking (LRIT) technique is mainly employed to identify ship positions when they sail in an ocean area (i.e., far away from the coastal region) [8–10]. However, the LRIT technique updates the ship position with a large time span (approximately every six hours), which cannot be used in real-time maritime applications.

AIS data are widely used to locate ship positions when they sail in inshore areas, and AIS-based ship positions are updated in a real-time manner. More specifically, the onboard AIS equipment broadcasts its position every 10 s. Note that the time interval can be smaller when the ship moves at a higher speed, and vice versa. AIS data have successfully fulfilled maritime situation awareness tasks (e.g., near-miss ship collision avoidance, ship trajectory exploitation and prediction, illegal maritime event identification, etc.) [11–15]. The following weaknesses may significantly reduce AIS performance: (1) it is not easy to collect fishery boat AIS data as it is not mandatory to be installed on the facility; (2) several ships may deliberately deactivate their AIS equipment for the purpose of avoiding being sensed by neighboring ships and maritime regulations (e.g., warships, smuggling ships, etc.). It is noted that we can employ object recognition techniques to further exploit distinct ship features from coastal maritime surveillance videos to identify warships and smuggling ships. (3) We cannot easily obtain noise-free AIS data due to fake plate ships, AIS data loss, etc. [16,17]. In that manner, many preprocessing steps (such as data quality control, anomaly detection, outlier correction, etc.) are required before employing AIS data to analyze maritime traffic situation awareness.

Computer vision-based techniques have enjoyed many successes when implemented for the ship-tracking task, showing potential in tackling the maritime traffic situation awareness problem under adverse weather constraints [18–20]. More specifically, visual ship tracking results provide microscopic on-spot traffic parameters which help maritime traffic participants (e.g., ship crew, maritime traffic authorities, etc.) make reasonable sailing decisions in advance. Chen et al. employed varied deep learning models (e.g., convolution neural network, you only look once (YOLO), etc.) to exploit high-level ship features, and thus maritime spatial-temporal information (e.g., trajectory, heading direction) could be extracted and further analyzed [21]. Zhang et al. introduced a discrete cosine transformation-based ship detection model for the purpose of efficient maritime traffic surveillance from non-stationary videos (e.g., cameras installed on buoys, ships, etc.) [22]. Jung et al. proposed a clustering-based optimization method to estimate ship positions from maritime video clips, which was tested in a realistic maritime rescue application in South Korea [23]. Chen et al. proposed a hybrid model via the help of a generative adversarial network and a convolutional neural network to accurately detect small ships in maritime surveillance videos [24]. Zhang et al. introduced a novel detection framework to detect ships in arbitrary orientations in the maritime images by developing a rotated regional proposed network [25]. Similar studies can be found in [26,27].

Previous studies suggest that visual ship tracking tasks are implemented by extracting visual features of the ship from maritime images (synthetic aperture radar (SAR), visible light-based videos, etc.), taken under relatively good weather conditions [28–30]. Much effort is put into exploiting hand-crafted features from maritime SAR images which are sensitive to sea clutter interference (i.e., they can be obviously observed under storm navigation environment) [31,32]. Deep learning relevant models have shown great success at the ship tracking task [33,34]. It is not easy to apply the deep learning-based models in the realistic maritime ship tracking tasks, due to the following two disadvantages: (1) we need to obtain sufficient ship training samples for the purpose of pre-training the model; (2) we need to deploy a powerful ship-borne computer to implement the deep learning relevant models on a ship. In our previous study, we introduced multi-view learning and sparse representation models to exploit distinct visual features of the ship from maritime video clips, which were further tested under typical maritime traffic scenarios [35]. To mitigate anomaly position oscillations, a data quality control procedure was integrated to obtain high-resolution ship positions from maritime images [36]. The above-mentioned models experienced satisfied tracking results with a large computational cost; however, they may not satisfy real-time ship tracking task challenges under different visibility weather conditions in real-world applications. Overall, previous ship tracking relevant studies can be roughly classified into three types by the manner of data sources (see Table 1). More specifically, relevant studies of AIS data employ ship position-related data to fulfill the ship

collision-avoidance task, and ship trajectory may be adjusted to avoid potential maritime traffic accidents. The radar relevant studies employ ship echoes to avoid potential ship accidents by performing ship maneuvering operations. Currently, more attention is being paid to extracting maritime traffic situations via the help of maritime surveillance images, and thus alerting the relevant officials of potentially dangerous traffic accidents in advance.

Table 1. Overview of the previous work related to ship tracking.

Data Type	Data Source	Target	Result
AIS	Ship position	Ship collision avoidance	Ship trajectory adjustment
Radar	Ship echoes	Inshore ship accident avoidance	Ship maneuvering operation
Maritime video	Maritime images	Visual traffic surveillance	Early-warning maritime traffic situation

The newly emerging kernelized correlation filter (KCF) models show satisfactory performance in many general-purpose tracking tasks (e.g., pedestrian tracking, vehicle localization, etc.) [37,38]. The conventional KCF model implements the object tracking task by exploring distinctive object features learning from both positive and negative training samples. In that manner, the KCF model can be considered as a potential method for fulfilling the real-time ship tracking task. It is noted that an intrinsic disadvantage for the relevant KCF models is their sensitivity to ship scale-variation interference (i.e., the KCF model can only tackle the scale-invariant tracking task). Ship imaging scales demonstrate the relevant position between different neighboring ships, and thus various types of ship kinematic information (e.g., speed, moving direction, etc.) can be inferred from maritime image sequences in advance. However, the classic KCF model can be easily interfered by the target ship scale variation during the ship tracking procedure, which can trigger a tracking drift phenomenon (i.e., obtain an obvious biased ship position). Moreover, ship tracking under different weather conditions (e.g., good visibility, bad visibility, etc.) is an important but on-going task. To address the above-mentioned issues, we introduce log-polar transformation to mitigate scale variation interference for the conventional KCF model (i.e., obtaining the ship scale factor). More specifically, the ship scale variation in the log-polar coordinate system can be easily observed in the form of offset in polar and angular coordinates (similar to the x and y axis in the Cartesian coordinate system), and thus it is easy to determine the ship scale factor variation in each maritime image. Based on the above analysis, we propose a scale-adaptive kernelized correlation filter (SKCF) to robustly track ships in real time under varied ship scale-interfered traffic scenarios at different weather conditions. We demonstrate our ship tracker performance in comparison with another two popular KCF-derivative trackers. We have provided a brief flowchart for the proposed framework of the study in Figure 1.

The remainder of the paper is organized as follows: Section 2 describes the proposed ship tracking model in detail. Section 3 introduces the dataset, evaluation metrics and the experimental results. Section 4 briefly concludes the research.

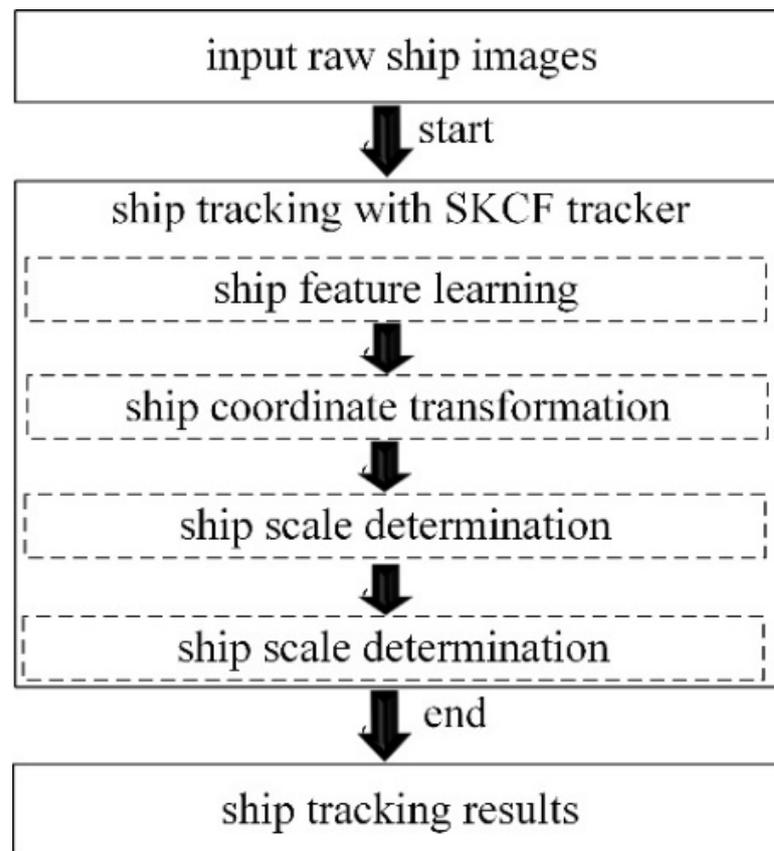


Figure 1. Flowchart of proposed SKCF tracker.

2. Methodology

2.1. Framework Overview

The framework for the proposed SKCF model is shown in Figure 2, which consists of ship position determination and ship scale refinement. More specifically, the SKCF model employs a conventional KCF method to obtain the ship's position in the raw image sequences by applying cosine windows to sample ship candidates on the Fourier-transformed ship images. Then, both the previous step output (i.e., raw ship tracking position) and ship training sample are transformed into log-polar coordinate systems. In that manner, the proposed tracker exploits the intrinsic ship contour feature with a histogram of oriented gradient (HOG) descriptor. Moreover, the ship scale factor is further identified by determining the offsets in both x and y directions in the log-polar coordinates. More specifically, the SKCF tracker identifies the ship tracking position and scale by finding the maximum response between the input training sample and KCF-obtained ship positions (i.e., potential candidates).

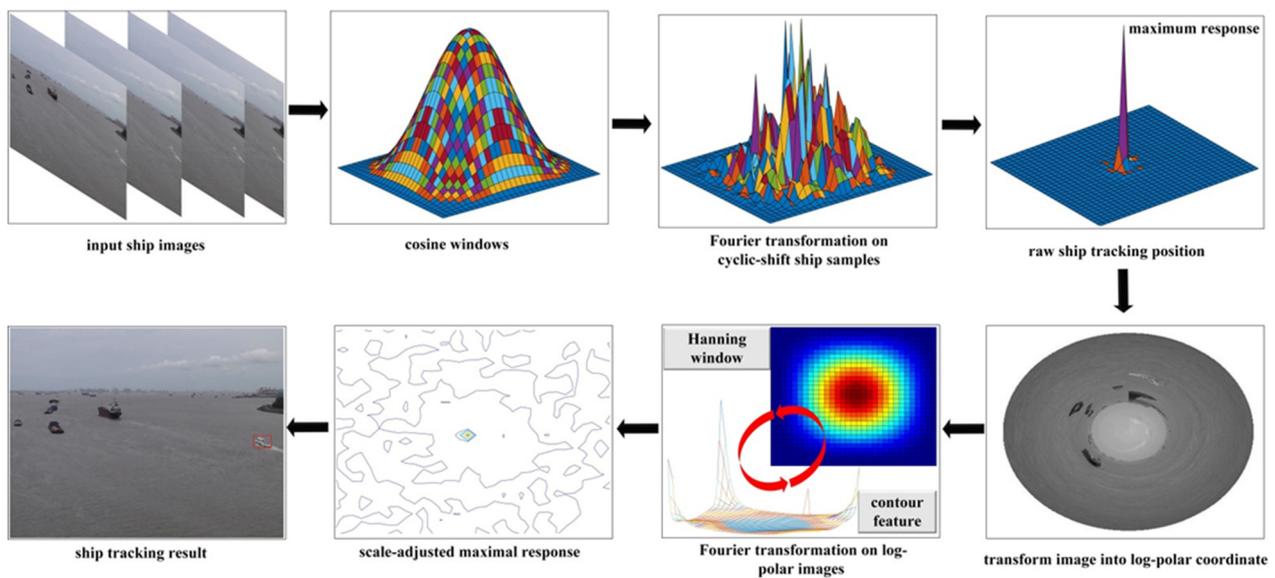


Figure 2. Working flow for the proposed scale-adaptive kernelized correlation filter.

2.2. Ship Tracking with KCF

The KCF model employs the cross-correlation criterion to identify the maximum similarity between the ship training sample and the ship candidate image [37]. More specifically, the KCF tracker identifies the ship position which shows a maximum stimulus response between the training and target ship frame in the Fourier domain. The KCF tracking output provides the initial ship tracking position in the current ship frame, which serves as the input for the ship scale refinement.

The ship training sample in frame k is presented in the form of (s_k, r_k) , and the ship tracker $f(s)$ is trained in an iterative manner. More specifically, the ship tracker $f(s)$ is the optimal solution for the Equation (1), which is indeed a linear regression problem. Previous studies suggest that the optimal solution for Equation (1) can be obtained by obtaining minimal distance between the ship sample and the candidate, which is reformulated as Equation (2). The close form solution (see Equation (3)) for Equation (2) is considered as an optimal solution. Note that the input ship training sample is transformed into feature space $\varphi(s_k)$ with kernel matrix b . The minimal solution for Equation (2) is considered as a linear addition of the ship training sample, and the ship tracking result (i.e., maximum response) obtained by the KCF model is formulated as Equation (4). In that manner, we re-formulate the KCF tracking model in form of Equation (5), and the ship position can be obtained via a closed form solution, as shown in Equation (6):

$$f(s) = \langle v, s \rangle + c \tag{1}$$

$$\min \sum_k L(f(s_k), r_k)^2 + \lambda \|v\|^2 \tag{2}$$

$$s = (p^T p + \lambda I)^{-1} p^T q \tag{3}$$

$$s = \sum_k \beta_k \varphi(s_k) \tag{4}$$

$$f(s) = \sum_k \beta_k b(s, s_k) \tag{5}$$

$$\beta = (B + \lambda I)^{-1} r \tag{6}$$

where the symbol $\langle \cdot, \cdot \rangle$ is dot production and parameter s is the linear combination of the ship training sample. The λ is the regularization magnitude of the trained ship tracker, and

symbol $L(f(s_k), r_k)^2$ is a loss function. The parameter B represents the kernel matrices, each of which contains a group of kernels b , and symbol I is the identity matrix. The parameter s consists of a group of s_k , which is applicable to the parameter r . The vector β is the optimal solution for the ship tracking model.

We employ a cyclic-shift mechanism to generate more training samples for the purpose of enhancing KCF tracker generalization capability (see Figure 3). More specifically, the cyclic-shift mechanism is iteratively applied on the ship base sample (i.e., the ground truth (GT)) to obtain ship samples, which may contain background pixels. The basic permutation matrix for acquiring the cyclic-shift ship samples is denoted as a dimensional vector $c = [c_1, c_2, \dots, c_m]$ (see Equation (7)). We can obtain a circulant kernel matrix $E(c)$ by cyclically sampling on the vector c , which can generate many ship base sample derivatives (see Equation (8)). We transform the circulant kernel matrix $E(c)$ into a diagonal matrix under the discrete Fourier transformation rule (see Equation (9)), where parameter F is the discrete Fourier transformation matrix. In that way, the ship training sample and labels are integrated into Equation (10). The symbol $*$ in Equation (10) is the complex-conjugate operator and \odot is the element-wise product.

$$n_s = \begin{bmatrix} 0 & 0 & \dots & 0 & 1 \\ 1 & 0 & \dots & 0 & 0 \\ 0 & 1 & \dots & 0 & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & \dots & 1 & 0 \end{bmatrix} \tag{7}$$

$$E(c) = \begin{bmatrix} c_1 & c_2 & c_3 & \dots & c_m \\ c_m & c_1 & c_2 & \dots & c_{m-1} \\ c_{m-1} & c_m & c_1 & \dots & c_{m-2} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ c_2 & c_3 & c_4 & \dots & c_1 \end{bmatrix} \tag{8}$$

$$E(c) = F \text{diag}(\hat{c}) F^B \tag{9}$$

$$\hat{v} = \frac{\hat{s}^* \odot \hat{r}}{\hat{s}^* \odot \hat{s} + \lambda} \tag{10}$$

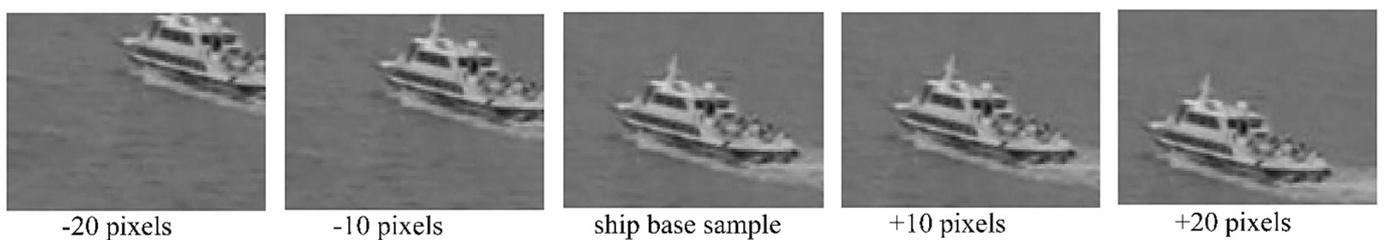


Figure 3. Ship training samples generated by the KCF cyclic-shift mechanism. The cyclic-shift ship samples are obtained by shifting starting point in both horizontal and vertical directions with certain space displacement (e.g., we minus 20 pixels from both x and y coordinates of the base ship sample without changing the width and length).

Note that obtaining the optimal solution for Equation (6) results in a spatial-temporal domain requiring large computational cost, which may fail to satisfy the real-time ship tracking task. Motivated by the success of our previous study [37], we transformed both ship training data (i.e., ship samples and candidates) into a frequency domain with the rule of the diagonalized Discrete Fourier Transform method. In that manner, the above-mentioned operators (e.g., multiplication, transposition, inversion) were applied in pixel-wise elements on each ship frame. For each ship frame, only half of the ship image would be searched for the purpose of finding the maximal response due to the advantage

of information redundancy of the diagonalized Discrete Fourier Transform (i.e., with a reduction in computational cost over 50%).

By defining a compact vector Υ (see Equation (11)), we reformulated the optimal KCF tracker solution (i.e., Equation (6)) as Equation (12), with the aim to efficiently obtain ship tracking results by suppressing the image redundancy information. Thus, the KCF tracking model considers the ship candidate area in the input maritime image by finding the maximal stimulus response between the trained KCF model and the input ship image. More specifically, the input ship image response in Equation (2) with minimal value is considered as the KCF tracking result, which is shown in Equation (13):

$$Y_m = Y(s, E(c)^k) \tag{11}$$

$$\beta = F^{-1}\left(\frac{F}{F(E) + \lambda}\right) \tag{12}$$

$$r' = \sum_k \beta_k \Gamma(p, C(b)^m) \tag{13}$$

where Y_m is a compact form of the kernel matrix $E(c)$. The division operator in Equation (12) is implemented in an element-wise manner, and the r' is the tracking result for the current ship image.

2.3. Ship Scale Refinement

The conventional KCF model may not function properly when the ship scale varies in the maritime video clips. More specifically, the KCF model may experience severe tracking outliers when the target ship length and/or width changes in each maritime image sequence, which is named as the tracking shifting phenomenon [39]. To overcome this issue, we introduced log-polar transformation to suppress the scale-variation interference considering the advantages of scale and rotation invariant properties. The log-polar image geometry theory suggests that scale and rotation variation in the Cartesian coordinate corresponds to the offset in the x and y axes in the log-polar coordinate system. We defined the polar coordinates (ρ, θ) which corresponded to radial distance and angle from the origin. Given the ship image center point (x_c, y_c) , we mapped each pixel (x, y) onto polar coordinates, which are shown as Equations (14) and (15). After that, the log-polar coordinates for the pixel (x, y) (from Cartesian coordinate) was reformulated as Equation (16). A mapping example for a ship image is shown in Figure 4.

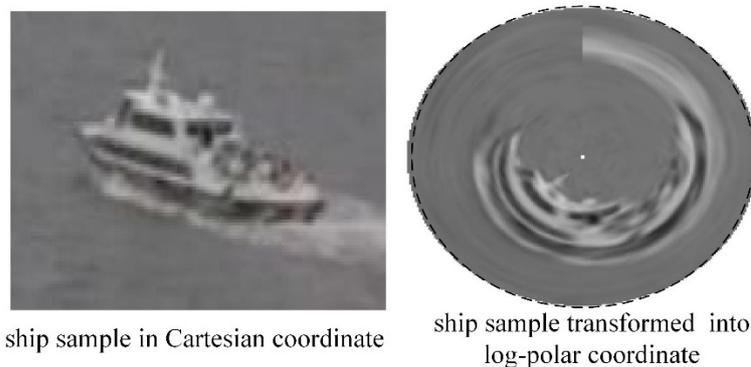


Figure 4. Ship sample mapping from Cartesian to log-polar coordinate.

Supposing the ship size in the current image is scaled with factor α , the ship coordinates (x, y) in the Cartesian domain will be transformed into $(\alpha x, \alpha y)$. In that manner, the corresponding coordinates in the log-polar domain for the ship coordinates were reformulated as Equation (17). We found that the scale variation in the Cartesian domain was transformed into a displacement from the polar (and angular) coordinates in the

log-polar domain. For the purposes of a robust and efficient tracking performance, the log-polar transformed ship image was transformed into the Fourier domain. Note that the common operations (e.g., translation, rotation, reflection, scale) in the Cartesian coordinate system are mapped onto their counterpart in Fourier domain [40,41]. Then, the Hanning window series were applied on the ship log-polar image transformed into the Fourier domain. The scale factor was determined with the phase correlation filter by identifying the maximal response. The pseudo code for the proposed SKCF ship tracking model is shown in Algorithm 1.

$$\rho = \sqrt{(x - x_c)^2 + (y - y_c)^2} \quad (14)$$

$$\theta = \arctan\left(\frac{y - y_c}{x - x_c}\right) \quad (15)$$

$$(\rho, \theta) = (\log(\rho), \theta) \quad (16)$$

$$(\log(\alpha x), \log(\alpha x)) = (\log(\alpha) + \log(x), \log(\alpha) + \log(y)) \quad (17)$$

Algorithm 1 The proposed SKCF ship tracker

Input: Ship images and ship position in the first frame.

Output: Estimated ship tracking position in current frame;

if the initial ship frame **then**

1. Performs parameter initialization;
2. Extracts pre-trained ship patterns and labels;
3. Trains the ship tracker in Fourier domain;

else

4. Extracts ship features from the previous image;
5. Transforms the ship image into Fourier domain;
6. Obtains maximal response and obtains raw ship position;
7. Transforms the ship (i.e., raw tracking result) and sample into the log-polar coordinate system;

8. Obtains maximal response in Fourier domain;

9. Determines ship scale factor;

end

3. Experiments

For the purpose of performance verification, we implemented our proposed SKCF ship tracker on typical maritime video clips, which contained a ship imaging scale variation challenge. The conventional KCF tracker and scale-adaptive tracker with multiple features (abbreviated as SAMF) were both tested on the same ship videos for the tracking performance comparison [37,42]. The three ship trackers were implemented on Windows 10 OS with the Intel Core CPU at 3.50 GHz processor, and the RAM was 8 G. Moreover, the GPU memory was 2 G (with NVIDIA GeForce GTX 850 M version). The ship tracking experiments were implemented on Matlab (2016 version). We carefully fine-tuned the parameters in our study to obtain a satisfactory model performance—the crucial parameter settings were as follows. The padding parameter was set to 1.3, and default patch size was set to 40. We tested the model performance with different settings for the learning rate, and it was found that the model could obtain a satisfactory performance when the learning rate was set to 0.003. Thus, we set the learning rate into 0.0003 in our study when no further specification was given in the study.

3.1. Data

We shot three maritime video clips for the purpose of evaluating ship tracker performance, denoted as video #1, #2, and #3. Video #1 aimed to test ship tracker performance when the ship scale in the image sequences showed a decreasing trend (i.e., ship imaging size became smaller in the latter frames). Video #1 contained 910 frames of images with a frame rate of 30 fps (i.e., frame per second) and an image resolution of 1280 × 720. Video

#2 was to test ship tracker performance when the target ship imaging size showed an increasing tendency. Moreover, the target ship in video #2 imposed the rotation challenge on the ship tracker. Video #2 included 700 frames taken at a frame rate of 30 fps. The image resolution for video #2 and #3 was same to that of video #1. We collected video #3 under poor visibility weather conditions for the purposes of testing the ship tracker performance under adverse weather conditions. Video #3 contained 600 frames, and both the frame rate and image resolution were same to the counterparts of video #1. More detailed information for the three maritime video clips are shown in Table 2, and the typical ship frames are shown in Figure 5. Note that the cameras were fixed onto on-shore buildings or ship rigid areas, and thus the issue of camera movement can be ignored in our study. We did not collect video clips under severe camera vibration, which was beyond our research focus. We will publish our collected maritime images in future, considering that a few maritime benchmarks containing ship scale variation challenges could be found.

Table 2. Detailed information for the three ship video clips.

Video No.	FRAME RATE	Resolution	Total Frame Number	Tracking Challenge
Video #1	30 fps	1280 × 720	910 frames	Ship size decreases in the video
Video #2	30 fps	1280 × 720	700 frames	Ship size increases in the video along with rotation challenge
Video #3	30 fps	1280 × 720	600 frames	Ship size decreases in the video taken in mid-foggy conditions



Figure 5. Ship frame samples (note that the red rectangle in each frame is the target ship).

3.2. Tracking Performance Measurements

To analyze ship tracker performance, we compared the distance between the tracked and ground truth ship positions. Note that the ground truth ship positions were manually labelled in a frame-by-frame manner. We measured the distance between the tracked and ground truth positions with the indicators of root mean square error (RMSE), mean absolute deviation (MAD), mean absolute percentage error (MAPE), and intersection over union (IOU) for the purpose of tracking model performance comparisons. Moreover, we employed the fps indicator to quantify computational cost. More specifically, we employed the center point to represent ship position in each maritime frame, which was further employed to calculate RMSE, MAD and MAPE. The IOU indicator represented the overlapping level between the tracked and ground truth ship position. Moreover, the computation cost was evaluated with the fps indicator. We considered that the ship tracker obtained real-time tracking performance when the fps was larger than 20 [43].

We evaluated varied tracker performances through the Euclidean distance between the center points between tracked and ground truth ship positions (i.e., each ship was presented by center point position). Given a maritime video clip, we employed the symbol $G(x,y)$ to represent the ground truth ship position, and $T(x,y)$ for the counterpart obtained by the ship tracker. The Euclidean distance between $G(x,y)$ and $T(x,y)$ was calculated by Equations (18)–(20). We can obtain the RMSE, MAD and MAPE distributions after obtaining average Euclidean distance (see Equations (21)–(24)). The smaller values for the above-mentioned four indicators showed higher tracking accuracy with less computation complexity, and vice versa. The IOU indicator (see Equation (25)) is a popular evaluation metric used in both object detection and tracking communities. According to the rule of thumb, we considered that the ship tracker obtained a satisfactory performance when the IOU score > 0.5 . The tracked ship position was well matched with the ground truth position (i.e., manually labeled ship position in image sequences) when the IOU score was 1.

$$N_n(x) = (G_n(x) - T_n(x))^2 \tag{18}$$

$$N_n(y) = (G_n(y) - T_n(y))^2 \tag{19}$$

$$N_n(G(x,y), T(x,y)) = \sqrt{N_n(x) + N_n(y)} \tag{20}$$

$$\bar{N} = \frac{\sum_{n=1}^k N_n(G(x,y), T(x,y))}{k} \tag{21}$$

$$RMSE = \sqrt{\frac{\sum_{n=1}^k |N_n(G(x,y), T(x,y)) - \bar{N}|^2}{k}} \tag{22}$$

$$MAD = \frac{\sum_{n=1}^k |N_n(G(x,y), T(x,y)) - \bar{N}|}{k} \tag{23}$$

$$MAPE = \frac{1}{k} \sum_{n=1}^k \left| \frac{N_n(G(x,y), T(x,y)) - \bar{N}}{N_n(G(x,y), T(x,y))} \right| \tag{24}$$

$$IOU = \frac{\text{Area of Overlap}}{\text{Area of Union}} \tag{25}$$

where $N_n(x)$ represents the displacement between ship position $G(x,y)$ and $T(x,y)$ on the x-coordinate at frame n. The total frame number in the ship video is k, and the corresponding x-coordinates for the two ship positions are denoted as $G_n(x)$ and $T_n(x)$, respectively. The rule is applicable to parameters $N_n(y)$, $G_n(y)$ and $T_n(y)$. The Euclidean distance between the two ship positions is represented as $N_n(G(x,y), T(x,y))$, and \bar{N} is the average Euclidean distance.

3.3. Ship Tracking Results on Video #1

For the purpose of representing the ship tracker performance, we obtained the displacement distribution between the tracked and GT positions in a frame-by-frame manner, abbreviated as D2TGT. The D2TGT distribution demonstrated the ship scale variation tendency impact on the conventional KCF model performance (see Figure 6). More specifically, the KCF model tracking error showed an increasing tendency, as the ship scale became smaller in the latter tracking image sequences. The D2TGT distribution shows obvious tracking anomaly since frame #350. After carefully checking the ship tracking result (by plotting the tracked ship positions on the video #1), we found that the KCF bounding box (i.e., tracked position) was significantly larger than that of the ground truth. The tracking outlier confirmed that the KCF tracking model could not handle the ship scale variation challenge. Compared with the KCF tracking performance, the SAMF and SKCF models showed less tracking errors (i.e., both the trackers obtained better tracking accuracy) as the D2TGT curves showed smooth variation tendency.

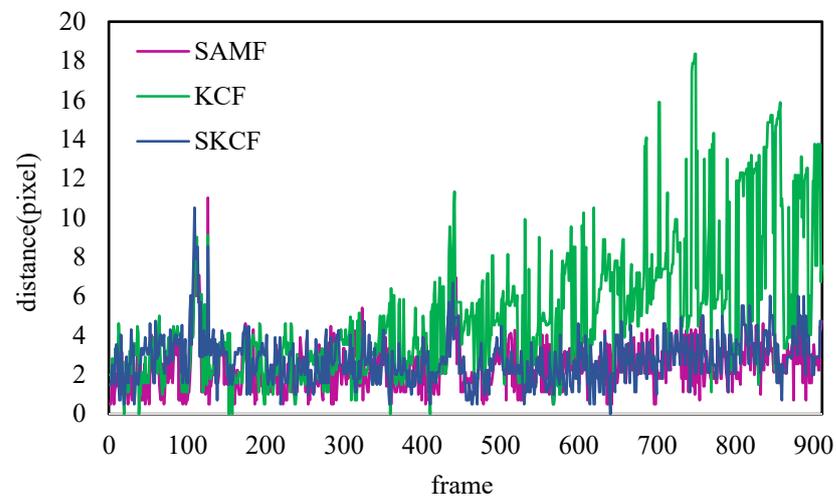


Figure 6. D2TGT curve distributions for the trackers on video #1.

We employed the RMSE, MAD, MAPE, IOU and fps to further compare the three trackers’ performance, which are shown in Table 3. The Euclidean distance-relevant metrics (i.e., RMSE, MAPE and MAD) showed that our proposed SKCF tracked ship position was closer to the ground truth than the other two models. More specifically, the RMSE for the KCF tracking model was 3.47 pixels, which is 2.9 times larger than that of the SKCF model. Note that the MAD obtained by the KCF model was also 2.9 times larger than the counterpart of the SKCF model. Similarly, the MAPE values for both the KCF and SAMF trackers were significantly larger than that of our model. The RMSE, MAD and MAPE for the proposed SKCF model were 1.19, 0.91 and 0.08—smaller than those of the KCF and SAMF models. From the perspective of Euclidean distance measurement, we can conclude that the proposed SKCF model obtained a satisfactory performance in video #1.

Table 3. Statistical metrics for varied ship trackers for video #1.

Model	RMSE	MAD	MAPE	IOU	fps
KCF	3.47	2.64	0.89	0.46	101.9
SAMF	1.22	0.92	0.11	0.73	5.4
SKCF	1.19	0.91	0.08	0.81	95

The IOU indicator presented a matched level between the tracked and ground truth ship positions, which provided a holistic evaluation on the tracker performance. The KCF model obtained an IOU score of 0.46 (smaller than the threshold 0.5), and thus we considered that the KCF tracker failed to track the ship in video #1. Note that the IOU

scores for both the SAMF and SKCF models were both larger than the threshold, which were 0.73 and 0.81, respectively. The IOU indicator suggests that our proposed ship tracker could obtain a better tracking accuracy. From the perspective of computational cost, both the KCF and SKCF models could obtain real-time tracking performance, and thereby the fps value for both trackers was larger than 90. The SAMF tracker obtained an fps of 5.4, indicating that the computational complexity was obvious larger than the two trackers. The main reason for this was that SAMF exploits several ship features (e.g., color, HOG, etc.) to fulfil the ship tracking task. Consequently, it is not easy to apply the SAMF tracker to implement resource-constrained maritime tasks (i.e., tracking ships with a ship-borne computer system). Considering the above analysis on the metrics about tracking accuracy and time cost, our proposed SKCF model obtained a better tracking performance than the other two models for video #1 (i.e., the SKCF obtained satisfactory ship tracking results in real-time).

For the purpose of visualizing the tracking performance of the three trackers, typical ship tracking frames are shown in Figure 7. Note that we have cropped the region of interest from the ship tracking image for the purposes of inspecting the ship tracking performance in detail. The three trackers' performances were very close (i.e., well-matched with the ground truth position) at the beginning of the tracking procedure (see the upper-left subplot in Figure 7). However, the KCF-tracked bounding boxes (i.e., tracked ship positions) were obviously larger than those of the ground truth in the latter three frames (where the ship scales became significantly smaller than that of frame #200). The bounding box gaps for the KCF tracker shown in Figure 7 confirmed our above analysis. Both the SAMF and SKCF trackers can automatically adjust the ship scale during the ship tracking procedure, as shown in the tracking results of frame # 400, #700 and #900 in Figure 7. The ship tracking results on the bottom-right subplot in Figure 7 show that the SAMF tracked position was larger than the ground truth, while our proposed ship tracker was much closer to the ground truth bounding box. The main reason was that the SAMF tracker may have been interfered by the color feature (mainly from the ship-wake pixels). Based on the above quantitative and qualitative analysis, we can conclude that our proposed SKCF tracker obtained a satisfactory performance during the tracking challenge for video #1.

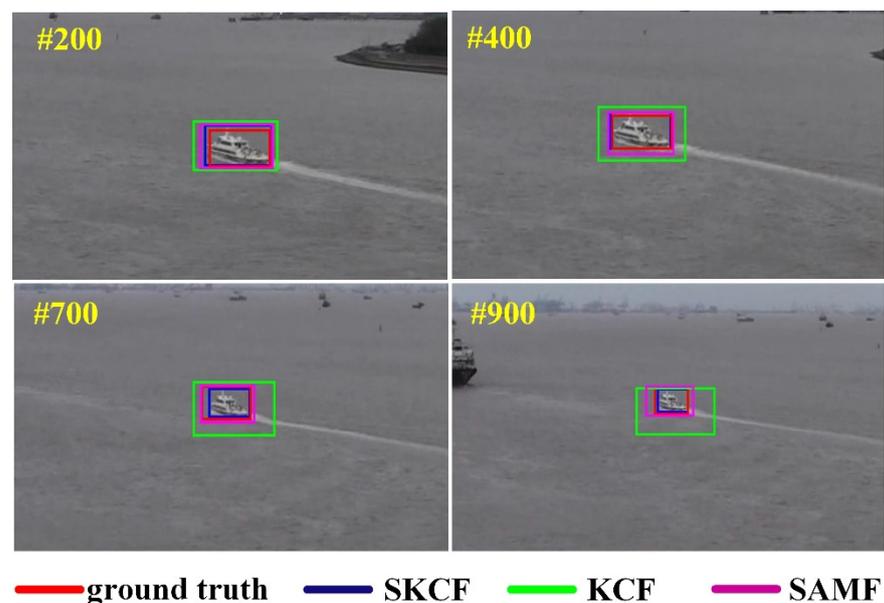


Figure 7. Ship tracker performance on typical frames for video #1.

3.4. Ship Tracking Results on Videos #2 and #3

The three tracking models were implemented on videos #2 and #3 to further test the tracker's robustness. We have not presented the typical frames with ship tracking results

due to page limitations. The D2TGT curve distributions for videos #2 and #3 are shown in Figures 8 and 9, respectively. The D2TGT curve distribution indicates that there was a negative correlation between the tracking performance of the KCF model and ship scale variation. More specifically, the KCF model obtained accurate ship tracking positions, as the Euclidean distance was smaller than 20 pixels (see the blue sample points from frame #1 to #500 in Figure 8, and frame #1 to #200 in Figure 9). The KCF Euclidean distance showed an obvious increasing variation tendency when the target ship decreased its imagine size (see the blue sample points from frame #500 to #700 in Figure 8, and frame #200 to #600 in Figure 9). In comparison to the KCF tracking results, the D2TGT curve distribution for SAMF was quite similar to that of the SKCF tracker. Overall, the Euclidian distance curves for the two trackers were very smooth, and only fluctuated at small magnitudes.

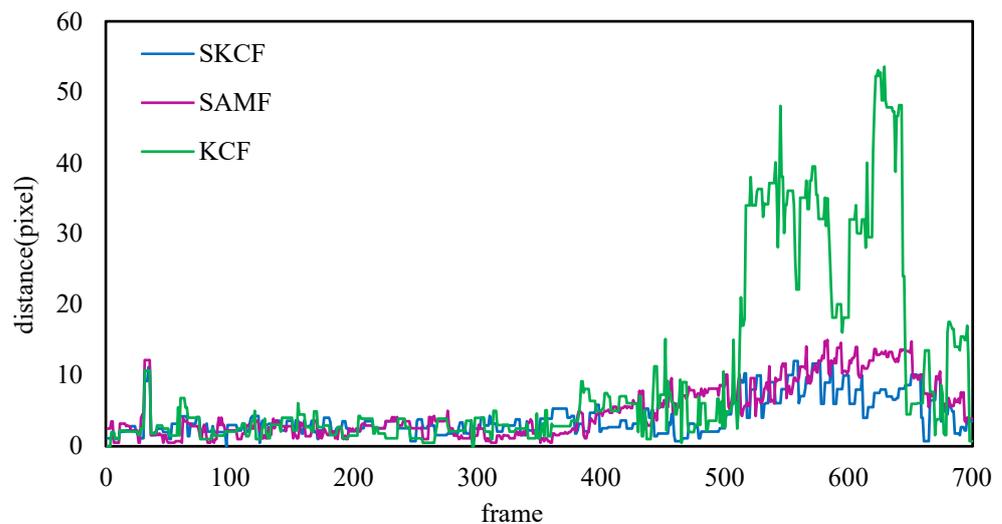


Figure 8. D2TGT curve distributions for the trackers on video #2.

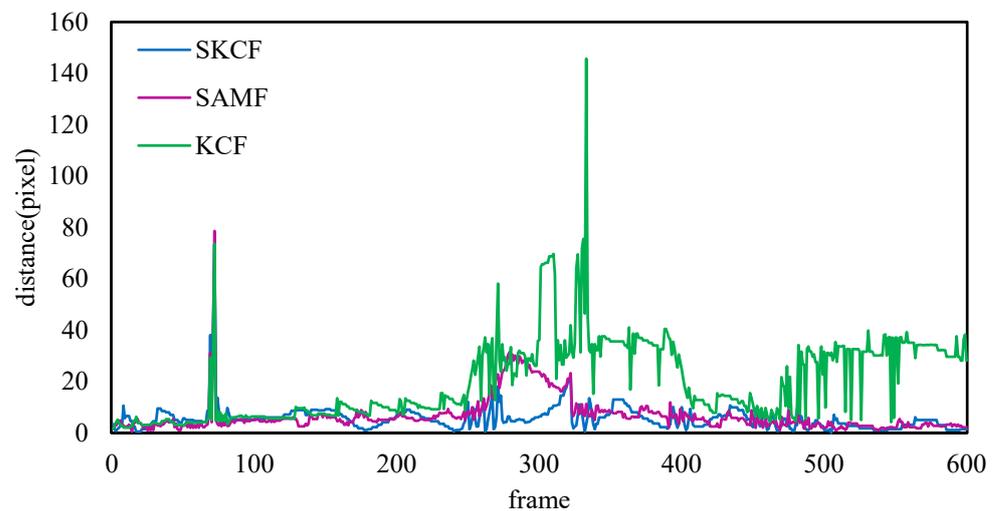


Figure 9. D2TGT curve distributions for the trackers on video #3.

The statistical metrics shown in Tables 4 and 5 further verify the three trackers’ performance in a quantitative manner. More specifically, the RMSE metric for the SKCF model was 2.66 (4.76) pixels in video #2 (#3), which is 4.83 (3.29) times to the counterpart of the KCF model of video #2 (#3). Meanwhile, the SAMF model obtained an RMSE of approximately 1.43 times greater than that of the SKCF model in both videos #2 and #3. The MAD indicator variation in Tables 4 and 5 showed a similar tendency as that of RMSE. The minimal MAD values for videos #2 and #3 were 2.08 and 2.79 pixels, respectively, which

were both obtained by the SKCF model. Moreover, the minimal MAPE values were also obtained by the proposed ship tracker, which are 0.73 and 1.01, respectively. From the perspective of IOU, the proposed SKCF model obtained a similar performance as that of the SAMF model in both videos #2 and #3, which were both higher than the KCF model counterparts. Note that the KCF model may have failed to exploit the ship scale variation tendency considering that the model focused on obtaining an accurate ship vertex position by exploring the spatial-temporal relationship between the ship candidate and template. In that way, the KCF distance was larger than the counterparts of the other two models. The fps statistical indicator suggested that only the KCF and SKCF models demonstrated the power of satisfying the real-time ship tracking task. Therefore, we can conclude that our proposed SKCF tracker successfully tracked ships in real time during the ship scale variation challenge of videos #2 and #3.

Table 4. Statistical metrics for varied ship trackers on video #2.

Model	RMSE	MAD	MAPE	IOU	fps
KCF	12.85	9.71	2.73	0.61	258.6
SAMF	3.81	3.19	1.38	0.63	6.2
SKCF	2.66	2.08	0.73	0.63	103.3

Table 5. Statistical metrics for varied ship trackers on video #3.

Model	RMSE	MAD	MAPE	IOU	fps
KCF	15.69	13.15	1.33	0.61	163.6
SAMF	6.75	4.06	0.87	0.81	13.9
SKCF	4.76	2.79	1.01	0.84	92.1

4. Conclusions

Visual ship tracking from maritime videos provides crucial microscopic traffic parameters for maritime situation awareness; this greatly benefits automatic navigation decision making in the smart ship era. We aimed to tackle the ship scale variation challenge to fulfill a holistic maritime situation exploitation task. More specifically, the proposed framework employed the KCF module to determine the initial ship position in the frame, and the ship scale factors were further refined with a log-polar-based kernelized correlation filter. We tested the proposed ship tracker performance under three video clips with typical maritime scenarios, and under different visibility conditions (i.e., sunny, fog). The statistics indicated that the proposed ship tracking model obtained a satisfactory performance during the ship scale variation challenge (in terms of robust tracking accuracy and low computational cost). More specifically, the average RMSE, MAD and MAPE were 2.87, 1.93 and 0.61, which indicated that the ship positions obtained by the proposed SKCF model were quite close to those of the ground truth data. In addition, the average IOU was 0.76 and the fps was 96.7, which confirmed our above analysis. The study can help on-duty ship officers in the ship bridge area obtain more informative kinematic traffic parameters in an easy yet convenient manner. Moreover, the study can further enhance maritime traffic safety by integrating with additional maritime traffic data sources (e.g., AIS). In this manner, more accurate ship maneuvering decisions can be made in advance, without requiring extra cost.

Though the proposed ship tracking model obtained a satisfactory performance under the scale variation challenge, we can still enhance the ship tracker performance in following directions. Firstly, we have already tested model performance on maritime traffic scenarios under good weather conditions without obvious sea clutter interference. We can further enhance our model performance by tackling challenges of adverse weather conditions (e.g., storm, thick fog), and thus promote our study to fulfil visual navigation tasks in the real-world smart ship era. Secondly, we have verified our model performance for tracking a single-ship target; verifying the tracker performance for a multiple-ship tracking task could be another interesting direction. Thirdly, we can integrate more efficient deep

learning models to address the ship scale variation challenge in terms of ship type detection. Fourth, we can enhance the performance of the ship tracking model under various coupled tracking challenges (e.g., bad visibility conditions, multi-dimensional camera movements) by combining different techniques (e.g., radar, LRIT).

Author Contributions: Conceptualization, H.L. and X.C.; methodology, H.L., X.X. and C.L.; writing—original draft preparation, H.L., X.X. and X.C.; writing—review and editing, C.L. and M.W.; funding acquisition, X.C. and C.L. All authors have read and agreed to the published version of the manuscript.

Funding: This work was jointly supported by National Key R & D Program of China (2019YFB1600602), National Natural Science Foundation of China (52102397, 52071200, 62176150), China Postdoctoral Science Foundation (2021M700790), Project of Zhoushan Science and Technology Bureau (No. 2021C21010), Fund of National Engineering Research Center for Water Transport Safety (No. A2022003), Foundation for Jiangsu key Laboratory of Traffic and Transportation Security (No. TTS2021-05).

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Readers can access our data by sending an email to the corresponding author Xinqiang Chen.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Yan, X.; Wang, K.; Yuan, Y.; Jiang, X.; Negenborn, R.R. Energy-efficient shipping: An application of big data analysis for optimizing engine speed of inland ships considering multiple environmental factors. *Ocean Eng.* **2018**, *169*, 457–468. [[CrossRef](#)]
2. Zhu, M.; Sun, W.; Hahn, A.; Wen, Y.; Xiao, C.; Tao, W. Adaptive modeling of maritime autonomous surface ships with uncertainty using a weighted LS-SVR robust to outliers. *Ocean Eng.* **2020**, *200*, 107053. [[CrossRef](#)]
3. Wu, B.; Cheng, T.; Yip, T.L.; Wang, Y. Fuzzy logic based dynamic decision-making system for intelligent navigation strategy within inland traffic separation schemes. *Ocean Eng.* **2020**, *197*, 106909. [[CrossRef](#)]
4. Zhou, F.; Pan, S.; Jiang, J. Verification of AIS Data by using Video Images taken by a UAV. *J. Navig.* **2019**, *72*, 1345–1358. [[CrossRef](#)]
5. Jiang, M.; Shen, J.; Kong, J.; Huo, H. Regularization Learning of Correlation Filter for Robust Visual Tracking. *IET Image Process.* **2018**, *12*, 1586–1594. [[CrossRef](#)]
6. Ma, W.; Hao, S.; Ma, D.; Wang, D.; Jin, S.; Qu, F. Scheduling decision model of liner shipping considering emission control areas regulations. *Appl. Ocean Res.* **2021**, *106*, 102416. [[CrossRef](#)]
7. Zhang, W.; Zou, Z.; Goerlandt, F.; Qi, Y.; Kujala, P. A multi-ship following model for icebreaker convoy operations in ice-covered waters. *Ocean Eng.* **2019**, *180*, 238–253. [[CrossRef](#)]
8. Ma, D.; Ma, W.; Jin, S.; Ma, X. Method for simultaneously optimizing ship route and speed with emission control areas. *Ocean Eng.* **2020**, *202*, 107170. [[CrossRef](#)]
9. Liu, R.W.; Liang, M.; Nie, J.; Lim, W.Y.B.; Zhang, Y.; Guizani, M. Deep Learning-Powered Vessel Trajectory Prediction for Improving Smart Traffic Services in Maritime Internet of Things. *IEEE Trans. Netw. Sci. Eng.* **2022**. Early access. [[CrossRef](#)]
10. Ma, D.; Ma, W.; Hao, S.; Jin, S.; Qu, F. Ship's response to low-sulfur regulations: From the perspective of route, speed and refueling strategy. *Comput. Ind. Eng.* **2021**, *155*, 107140. [[CrossRef](#)]
11. Fang, Z.; Yu, H.; Ke, R.; Shaw, S.-L.; Peng, G. Automatic Identification System-Based Approach for Assessing the Near-Miss Collision Risk Dynamics of Ships in Ports. *IEEE Trans. Intell. Transp. Syst.* **2019**, *20*, 534–543. [[CrossRef](#)]
12. Chen, Z.; Xue, J.; Wu, C.; Qin, L.; Liu, L.; Cheng, X. Classification of vessel motion pattern in inland waterways based on Automatic Identification System. *Ocean Eng.* **2018**, *161*, 69–76. [[CrossRef](#)]
13. Yang, D.; Wu, L.; Wang, S.; Jia, H.; Li, K.X. How big data enriches maritime research—a critical review of Automatic Identification System (AIS) data applications. *Transp. Rev.* **2019**, *39*, 755–773. [[CrossRef](#)]
14. Hörteborn, A.; Ringsberg, J.W.; Svanberg, M.; Holm, H. A Revisit of the Definition of the Ship Domain based on AIS Analysis. *J. Navig.* **2019**, *72*, 777–794. [[CrossRef](#)]
15. Ma, W.; Lu, T.; Ma, D.; Wang, D.; Qu, F. Ship route and speed multi-objective optimization considering weather conditions and emission control area regulations. *Marit. Policy Manag.* **2020**, *48*, 1053–1068. [[CrossRef](#)]
16. Zhao, L.; Shi, G. Maritime Anomaly Detection using Density-based Clustering and Recurrent Neural Network. *J. Navig.* **2019**, *72*, 894–916. [[CrossRef](#)]
17. Wu, B.; Yip, T.L.; Yan, X.; Soares, C.G. Review of techniques and challenges of human and organizational factors analysis in maritime transportation. *Reliab. Eng. Syst. Saf.* **2021**, *219*, 108249. [[CrossRef](#)]
18. Prasad, D.K.; Dong, H.; Rajan, D.; Quek, C. Are Object Detection Assessment Criteria Ready for Maritime Computer Vision? *IEEE Trans. Intell. Transp. Syst.* **2020**, *21*, 5295–5304. [[CrossRef](#)]

19. Prasad, D.K.; Prasath, C.K.; Rajan, D.; Rachmawati, L.; Rajabally, E.; Quek, C. Object Detection in a Maritime Environment: Performance Evaluation of Background Subtraction Methods. *IEEE Trans. Intell. Transp. Syst.* **2018**, *20*, 1787–1802. [[CrossRef](#)]
20. Kong, J.; Wang, B.; Jiang, M. Robust part-based visual tracking via adaptive collaborative modelling. *IET Image Process.* **2019**, *13*, 1648–1657. [[CrossRef](#)]
21. Chen, X.; Qi, L.; Yang, Y.; Luo, Q.; Postolache, O.; Tang, J.; Wu, H. Video-Based Detection Infrastructure Enhancement for Automated Ship Recognition and Behavior Analysis. *J. Adv. Transp.* **2020**, *2020*, 7194342. [[CrossRef](#)]
22. Zhang, Y.; Li, Q.-Z.; Zang, F.-N. Ship detection for visual maritime surveillance from non-stationary platforms. *Ocean Eng.* **2017**, *141*, 53–63. [[CrossRef](#)]
23. Jung, C.-Y.; Yoo, S.-L. Optimal Rescue Ship Locations Using Image Processing and Clustering. *Symmetry* **2019**, *11*, 32. [[CrossRef](#)]
24. Chen, Z.; Chen, D.; Zhang, Y.; Cheng, X.; Zhang, M.; Wu, C. Deep learning for autonomous ship-oriented small ship detection. *Saf. Sci.* **2020**, *130*, 104812. [[CrossRef](#)]
25. Zhang, Z.; Guo, W.; Zhu, S.; Yu, W. Toward Arbitrary-Oriented Ship Detection With Rotated Region Proposal and Discrimination Networks. *IEEE Geosci. Remote Sens. Lett.* **2018**, *15*, 1745–1749. [[CrossRef](#)]
26. Zhao, Y.; Zhao, L.; Xiong, B.; Kuang, G. Attention Receptive Pyramid Network for Ship Detection in SAR Images. *IEEE J. Sel. Top. Appl. Earth Obs. Remote Sens.* **2020**, *13*, 2738–2756. [[CrossRef](#)]
27. Wang, Y.; Wang, C.; Zhang, H.; Dong, Y.; Wei, S. A SAR Dataset of Ship Detection for Deep Learning under Complex Backgrounds. *Remote Sens.* **2019**, *11*, 765. [[CrossRef](#)]
28. Biondi, F. Low-Rank Plus Sparse Decomposition and Localized Radon Transform for Ship-Wake Detection in Synthetic Aperture Radar Images. *IEEE Geosci. Remote Sens. Lett.* **2017**, *15*, 117–121. [[CrossRef](#)]
29. Liu, T.; Zhang, J.; Gao, G.; Yang, J.; Marino, A. CFAR Ship Detection in Polarimetric Synthetic Aperture Radar Images Based on Whitening Filter. *IEEE Trans. Geosci. Remote Sens.* **2020**, *58*, 58–81. [[CrossRef](#)]
30. Yuan, D.; Kang, W.; He, Z. Robust visual tracking with correlation filters and metric learning. *Knowl. Based Syst.* **2020**, *195*, 105697. [[CrossRef](#)]
31. Filippo, B. COSMO-SkyMed staring spotlight SAR data for micro-motion and inclination angle estimation of ships by pixel tracking and convex optimization. *Remote Sens.* **2019**, *11*, 766. [[CrossRef](#)]
32. Gao, G.; Huang, K.; Gao, S.; He, J.; Zhang, X. Ship Detection Based on Oceanic Displaced Phase Center Antenna Technique in Along-Track Interferometric SAR. *IEEE J. Sel. Top. Appl. Earth Obs. Remote Sens.* **2019**, *12*, 788–802. [[CrossRef](#)]
33. Yuan, D.; Fan, N.; He, Z. Learning target-focusing convolutional regression model for visual object tracking. *Knowl. Based Syst.* **2020**, *194*, 105526. [[CrossRef](#)]
34. Yuan, D.; Li, X.; He, Z.; Liu, Q.; Lu, S. Visual object tracking with adaptive structural convolutional network. *Knowl. Based Syst.* **2020**, *194*, 105554. [[CrossRef](#)]
35. Liu, R.W.; Yuan, W.; Chen, X.; Lu, Y. An enhanced CNN-enabled learning method for promoting ship detection in maritime surveillance system. *Ocean Eng.* **2021**, *235*, 109435. [[CrossRef](#)]
36. Benetazzo, A.; Barbariol, F.; Bergamasco, F.; Torsello, A.; Carniel, S.; Sclavo, M. Stereo wave imaging from moving vessels: Practical use and applications. *Coast. Eng.* **2016**, *109*, 114–127. [[CrossRef](#)]
37. Henriques, J.F.; Caseiro, R.; Martins, P.; Batista, J. High-Speed Tracking with Kernelized Correlation Filters. *IEEE Trans. Pattern Anal. Mach. Intell.* **2014**, *37*, 583–596. [[CrossRef](#)]
38. Liu, Q.; Li, X.; He, Z.; Fan, N.; Yuan, D.; Wang, H. Learning Deep Multi-Level Similarity for Thermal Infrared Object Tracking. *IEEE Trans. Multimed.* **2020**, *23*, 2114–2126. [[CrossRef](#)]
39. Max, X.; Liu, X.; Li, Y. Fast Scale-Adaptive Correlation Tracking. *J. Comput. Aided Des. Comput. Graph.* **2017**, *29*, 450–458.
40. Zokai, S.; Wolberg, G. Image registration using log-polar mappings for recovery of large-scale similarity and projective transformations. *IEEE Trans. Image Process.* **2005**, *14*, 1422–1434. [[CrossRef](#)]
41. Sarvaiya, J.; Patnaik, S.; Kothari, K. Image Registration Using Log Polar Transform and Phase Correlation to Recover Higher Scale. *J. Pattern Recognit. Res.* **2012**, *7*, 90–105. [[CrossRef](#)]
42. Li, Y.; Zhu, J. A scale adaptive kernel correlation filter tracker with feature integration. In *European Conference on Computer Vision*; Springer: Berlin/Heidelberg, Germany, 2014.
43. Chen, X.; Ling, J.; Wang, S.; Yang, Y.; Luo, L.; Yan, Y. Ship detection from coastal surveillance videos via an ensemble Canny-Gaussian-morphology framework. *J. Navig.* **2021**, *74*, 1252–1266. [[CrossRef](#)]