A Robust Deep Affinity Network for Multiple Ship Tracking

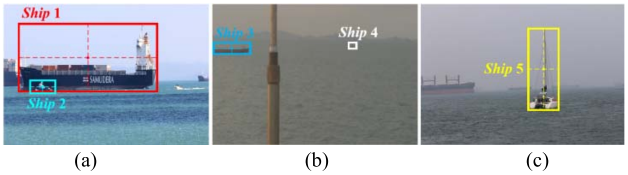
Wen Zhang, Xujie He, Wanyi Li, Zhi Zhang, Yongkang Luo, Li Su, and Peng Wang, *Member, IEEE*

***Abstract*— Multiple ship tracking (MST) is an important task in marine surveillance and ship situational awareness systems. Considerable work has been conducted on multiple object tracking in recent years, but it has focused primarily on pedestrians and automobiles, leaving a gap in studies on MST due to the particularities of complex marine scenes, such as ship scale variations, the long-tailed distribution of ships, and long-term occlusions caused by ship movements. In this article, we present a robust deep affinity network (RoDAN) for MST. To overcome the above difficulties in MST, we start with the basic deep affinity network (DAN) and improve it in three aspects: scale, region, and motion. For the scale dimension, we integrate an atrous spatial pyramid pooling (ASPP) module to improve the modeling ability for multiscale ships. For the region dimension, we propose the joint global region modeling (JGRM) module, which further strengthens the modeling ability of DAN and exploit it to overcome the long-tailed distribution property of ships. For the motion dimension, we propose the motion-matching optimization (MMO) module to fine-tune the tracking results and make our tracker more robust, less reliant on the front-end detector, and ameliorate long-term occlusions. The experimental results demonstrate that our MST method outperforms the state-of-the-art methods. In particular, it reduces the number of ID switches (IDSs) and trajectory fragmentations (FMs), achieving holistically preferable performance. Meanwhile, our method achieves a comparable speed.**

***Index Terms*— Complex marine scenes, joint global region modeling (JGRM) module, marine surveillance, motion-matching optimization (MMO) module, multiple ship tracking (MST).**

I. Introduction

SHIP driver misjudgments of the locations and behaviors of surrounding ships is a key factor leading to water (ocean/canal) traffic accidents. Automated ship locating and ship tracking mechanisms can not only help avoid human errors in observation but also provide a basis for analyzing the sailing behaviors of surrounding ships. Thus, information from such automated systems can be used as supplementary information to aid in ship navigation and avoid ship collisions. Multiple ship tracking (MST), which relies on computer vision and image processing techniques, can automatically locate and track surrounding ships. Compared to the automatic identification systems (AISs) currently used in ship navigation systems, vision-based MST has several advantages, including active data acquisition (no data can be acquired if an AIS is not installed or turned on), virtually no need to rely on ship driver manipulations, and more reliable location information. Therefore, MST plays an increasingly important role in marine surveillance and ship situational awareness systems. Furthermore, in marine scenes, with the goal of satisfying the demands of upper-level tasks, such as autonomous ship driving, much more attention should be paid to the continuous tracking capability in MST; in other words, the ID switch (IDS) metric is most focused on in MST.



(a) (b) (c)

Fig. 1. Challenges to ship tracking: (a) scale variation caused by ship types: *Ship* 1 and *Ship* 2 are two different types of ships with widely varying sizes; *Ship* 1 in (a), *Ship* 3 in (b), and *Ship* 5 in (c) all illustrate the long-tailed distribution property of ships, because these ships tend to occupy half or less of each predetected ship region. *Ship* 4 in (b) appears tiny due to its distance from the camera.

Compared to automobile and pedestrian tracking, complex marine scenes introduce several other challenges to ship tracking, including:

1) *Large-scale variation among ships:* ship scales vary among different types of ships [1], [2] and their sizes vary with the distance to the camera. For instance, oil tankers are usually much larger than are jet skis, as shown in Fig. 1(a), *Ship* 1 and *Ship* 2. Similarly, near-offshore ships usually appear larger than do far-offshore ships in visual images due to the vast sizes of marine scenes, as shown in Fig. 1(b), *Ship* 3 and *Ship* 4.
2) *The long-tailed distribution within predetected ship regions:* ship masts and superstructures give the shapes of ships a long-tailed distribution property. As shown in Fig. 1(a)–(c), the backgrounds of the *Ship* 1, *Ship* 3, and *Ship* 5 regions usually occupy a larger space when

Manuscript received April 12, 2021; accepted April 23, 2021. Date of publication May 5, 2021; date of current version May 19, 2021. This work was supported in part by the Development Project of Ship Situational Intelligent Awareness System under Grant MC-201920-X01 and in part by the National Natural Science Foundation of China under Grant 61771471. The Associate Editor coordinating the review process for this article was Dr. Lihui Peng. *(Corresponding author: Xujie He.)*

Wen Zhang, Xujie He, Zhi Zhang, and Li Su are with the Key Laboratory of Intelligent Technology and Application of Marine Equipment, Ministry of Education, Harbin Engineering University, Harbin 150001, China, and also with the College of Intelligent Systems Science and Engineering, Harbin Engineering University, Harbin 150001, China (e-mail: zhangwen@hrbeu.edu.cn; hexujie@hrbeu.edu.cn; zhangzhi1981@hrbeu.edu.cn; suli406@hrbeu.edu.cn).

Wanyi Li, Yongkang Luo, and Peng Wang are with the Institute of Automation, Chinese Academy of Sciences, Beijing 100190, China (e-mail: wanyi.li@ia.ac.cn; yongkang.luo@ia.ac.cn; peng_wang@ia.ac.cn).

Digital Object Identifier 10.1109/TIM.2021.3077679

ships are larger or equipped with masts and superstructures, which is challenging for ship tracking.

3) *Long-term occlusions:* some types of ships, such as very large ore carriers (VLOCs), tend to move at low speeds, which can cause long-term occlusions of other ships.
4) Other challenges caused by weather can also impact the tracking performance, such as illumination variations caused by clouds, poor visibility caused by sea fog, and reflections (glints) [3].

Because little research has been conducted on MST in recent years, ship tracking is still in an early development stages when compared to pedestrian or automobile tracking. Due to the limitations and particularities of marine backgrounds, research on MST also remains limited, because attention has been focused on more common types (e.g., pedestrians, automobiles) of multiple object tracking (MOT). Regarding MOT, clearly neural network-based trackers dominate other relevant theory-based trackers due to their powerful feature extraction abilities, which has been demonstrated in many fields [4], [5]. Neural network-based MOT (NN-MOT) methods can be categorized into three distinct groups: 1) detection-free tracking (DFT) [6]–[8], which requires manual initialization of objects in the first frame and then tracking them in subsequent frames without detection [9]; 2) detection-joint tracking (DJT) [10]–[14], which integrates object detection and object tracking into a single model to accomplish the tracking task; and 3) detection-based tracking (DBT) [15]–[23], in which all relevant objects are first detected in each individual frame by exploiting various off-the-shelf object detectors and then constructs one-to-one association relationships to link the detected objects in the current frame to objects in preceding frames. DJT and DBT have become popular approaches due to their flexibility. However, in marine scenes, ship detection is provided by marine surveillance systems or by smart ships; therefore, we need to focus only on addressing the association problem, making DBT the most suitable approach for marine scenes.

Despite achieving good tracking precision, NN-MOT methods still possess some disadvantages for tracking multiple ships.

1) Large-scale variations. NN-MOT methods usually concentrate less on scale issues because the objects they typically track (e.g., pedestrians, automobiles) do not usually exhibit large-scale variations. However, the situation is completely different when tracking ships, because the large-scale variations of ships degrade the accuracy of the affinity similarity that the network outputs, which results in ID switching. Thus, considering multiscale tracking is critical in MST.
2) Long-tailed distribution. NN-MOT methods have certain limitations when faced with objects with a long-tailed distribution: pedestrians and automobiles tend to occupy the complete predetected region, while ships do not. In this case, NN-MOT methods are less robust at modeling the features of all ships, resulting in IDSs.
3) Long-term occlusions. Occlusions that occur in most MOT situations usually last for only a few frames. However, due to the slow speeds of ships, occlusions in MST tend to last much longer, resulting in the long-term disappearances of ships. Moreover, DBT methods all place undue reliance on detectors; in other words, a better detector brews a better tracker. For example, consider a ship that appears in the current frame but is not detected by the front-end detector. In this case, this ship will be regarded as a "lost" ship by the back-end tracker, and it will be unable to conduct association operations with previous frames. Thus, making multiple ship trackers less reliant on detectors is vital. These issues explain why the current NN-MOT methods are not completely suitable for complex marine scenes.

In this article, to overcome the above special issues with marine scenes, we propose a method for MST based on a deep affinity network (DAN) [15], which fuses scale and motion as two separable dimensions. Moreover, to strengthen the modeling ability of DAN, we fuse the region dimension as another separable dimension. To summarize, the contributions of our work are as follows.

1) We propose a robust deep affinity network (RoDAN) for MST that fuses the discriminative information from three separable dimensions: scale, region, and motion.
2) To improve tracker robustness, less reliant on detectors, and ameliorate long-term occlusions, a motion-matching optimization (MMO) module that employs the ship's motion property to fine-tune the tracking results is proposed. In addition, to overcome the long-tailed distribution property, a joint global region modeling (JGRM) module that strengthens the modeling ability of DAN for feature expression is proposed. Finally, to address the scale variations of ships, an atrous spatial pyramid pooling (ASPP) module is adopted.
3) To encourage future research on MST, we have collected and annotated a new marine dataset named the Harbin Engineering University (HEU) ShipTrack Dataset (HSD). This dataset contains more challenging scenarios that often occur when tracking ships, such as heavy sea fog, long-term occlusions, and camera jitter. We plan to make this dataset publicly available to the scientific community.
4) We validate the effectiveness of our method on the Singapore Maritime Dataset (SMD) and our HSD and include thorough ablation studies. Compared with the state-of-the-art methods, our proposed method achieves better results with regard to IDSs and trajectory fragmentation (FM).

The remainder of this article is organized as follows. In Section II, we review the existing literature on MST and MOT. In Section III, we provide the details of the RoDAN for MST. In Section IV, we report the experimental and comparison results and discuss them in Section V. Finally, in Section VI, we draw conclusions and suggest possible future work.

II. Related Work

Previous studies on ship tracking can be categorized into four groups, each of which relies on a different methodology:

1) Background subtraction-based methods. Frost and Tapamo [1] and Szpak and Tapamo [2] utilize background subtraction to locate all the ships that appear in an image. This method works well in some specific scenarios: such as in calm wave conditions. However, this method requires one prerequisite: it must assume that the ships all move dynamically throughout the entire video; otherwise, static ships would be regarded as part of the background. Furthermore, when facing strong wave scenarios, the tracking accuracy degrades substantially because waves are dynamic and can be easily mistaken for moving ships.
2) Level set-based methods consider ship contours as important discriminative information. In [2], level set was used to extract the contours of all ships for tracking as supplementary information. Unfortunately, this approach loses its robustness when many glints exist on the water surface.
3) Kalman filter-based methods. Many ship trackers utilize the Kalman filter [24], [25] or its derivative forms, such as the extended Kalman filter [26], [27], to track ships. The Kalman filter requires fewer computing resources and is less complex than other methods. However, the key problem in utilizing the Kalman filter is that the environmental noise is unknown and changes constantly, which results in noise covariance deviations.
4) Correlation filter-based methods [28], [29] utilize a correlation filter to accomplish the task of ship tracking because this approach can be used to construct high-speed trackers, which is beneficial for industrial applications. However, experiments [30] show that correlation filter-based trackers are less robust to ship scale changes and can result in drift when occlusions occur. Another problem that cannot be neglected when utilizing the correlation filter method, is the boundary effect [28], which is also challenging to multifeature fusion-based tracking methods. The boundary effect reflects the fact that single features cannot adequately express each ship; instead, [31] proposes a multiview and sparse representation method that jointly utilizes edge and contour information. Other fusion methods, such as fusing features from electro-optical (EO)/IR images [32], satellite data, and AIS data [33], are also novel approaches to ship tracking.

Previous studies exploited for ship tracking can also be categorized into two sets relying on the number of ships when tracking: 1) single ship tracking (SIT) [5]–[29], [31], [32] and 2) MST [1]–[26], [33]. It appears that more attention is given to SIT since it is commonly regarded that MST can be approached with different independent SIT trackers. However, [24] finds that multiple independent SIT trackers are not ideal in tackling occlusions and interactions among different ships. At this point, it is of great necessity to design a tracker to robustly track multiple ships.

Research on MST remains limited compared to pedestrian tracking and automobile tracking. Aiming at determining a new direction for the MST problem, grasping the development process of MOT is of great necessity. To the best of our knowledge, widely used methods for MOT include seven main types: foreground modeling-based MOT methods [34]–[40], dynamic programming-based MOT methods [41], optical flow-based MOT methods [42]–[47], clustering-based MOT methods [48]–[51], Markov random field-based MOT methods [36]–[38], conventional machine learning-based MOT methods [52]–[59], and NN-MOT methods [15]–[23]. As mentioned above, NN-MOT methods stand out among other relevant theory-based MOT methods because of their powerful feature extraction ability. An increasing number of NN-MOT methods have been put forward into the literature in recent years. For instance, [16] adopted a topologically formed Siamese [60] network, exploited two same-sized image patches as the input, and output the corresponding matching similarity to obtain the final tracking results. Tang *et al.* [17] first introduced deep matching, which exploited a deep learning framework to calculate optical flow features and achieved promising tracking results. Inspired by Wojke *et al.* [18] and Bewley *et al.* [25] integrated appearance features using a neural network, which enormously reduced the number of IDSs. By analyzing the tracking scenes, [19] found that occlusion was the key point leading to drift and proposed the spatial-temporal attention mechanism to solve the occlusion problem for tracking.

Considering that MOT requires historical trajectories to predict current states, [20] designed a long–short-term memory (LSTM)-based feature fusion (FF) network to calculate the similarity between historical trajectories and current detections. Kim *et al.* [21] introduced a neural gating network using a bilinear LSTM to solve the deficiency in [20], and [22] proposed a novel multimodel and multicue (M3C) pipeline that exploited the gated recurrent unit (GRU) + Attention model fusion of maneuver, appearance, and pose information to enhance the detection and tracking performance. Zhang *et al.* [23] proposed a fusion network that fused image and point cloud features captured from different modalities to improve tracker reliability and accuracy. Nonetheless, all of these NN-MOT methods exploit neural networks to merely obtain object features (object representation); therefore, they all follow the same paradigm: exploiting two distinct models to accomplish the object representation and object association tasks. In other words, the association model must be tailored to the representation model.

Sun *et al.* [15] creatively proposed an end-to-end DAN that directly output the similarity of two objects by drawing on a single model within a single network. DAN has achieved good tracking performances in many scenarios. The emergence of DAN caused researchers to further rethink the solutions to the association problem. This capability is the first reason we chose DAN as our backbone. The second reason is because DAN has the ability to retrack occluded or out-of-view objects when they reappear in images, which is always another thorny challenge in ship tracking [29]. However, when we tested DAN for ship tracking, the issue shown in Fig. 2 occurred—IDSs caused by scale variations. Interestingly, as shown in Fig. 2, row 4, the ID could even switch to a nonship class (person), which indicates that the problem involves not only scale variation but also semantic issues that can impact ship tracking. Therefore, to improve the scale-agnostic and semantic abilities of our tracker, we adopted a multiscale fusion module called

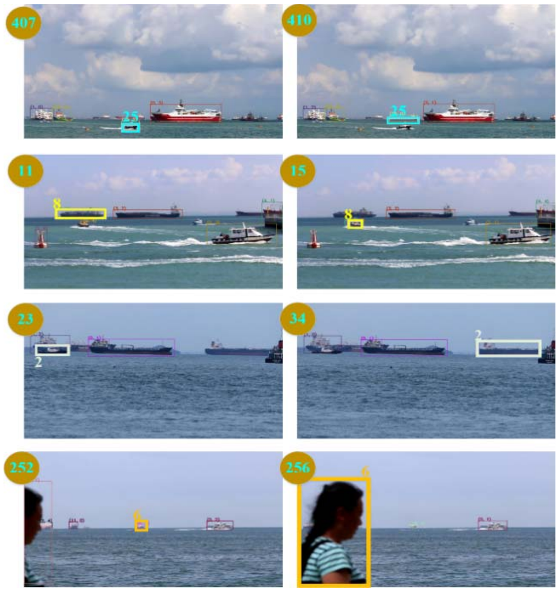


Fig. 2. Tracking results when exploiting DAN. The number in the upper left corner of each picture represents the frame index, and the two pictures in each row are captured from the identical video. As can be observed from the first, second, and third rows, the ID in the previous frames all switches to another ship with different scales in the next frame. In the fourth row, ID **6** even switches to an object of a nonship class (person).

ASPP to extract holistic semantic information from different receptive fields under different feature scales. We use ASPP to improve the appearance modeling ability for multiscale ships in the scale dimension. Next, we exploit a JGRM module for the region dimension to strengthen DAN's modeling ability, because the original DAN utilizes only center feature points in each predetected ship region as each ship's feature. As explained above, ships possess a long-tailed distribution property; therefore, utilizing only center points to express each ship is improper, because the center points of ship regions tend to be located in the background at times. Additionally, to make our tracker more robust and less reliant on the front-end detector and to ameliorate long-term occlusion problems, we exploit a MMO module that draws on ships' motion properties at the back end to fine-tune the tracking results in the motion dimension. Compared with previous works, a key peculiarity of our method is that we propose a multidimensional network that fuses scale, region, and motion dimensions to track ships. The advantage of our method is that our tracker is more suitable for MST under complex marine scenes. In particular, it reduces the IDS and FM problems, which indicates that our tracker can stably and uninterruptedly track multiple ships.

III. Robust DAN

The proposed RoDAN contributes three modules based on DAN to holistically improve the performance for MST: 1) ASPP module for the scale dimension; 2) the JGRM module for the region dimension; and 3) the MMO module for the motion dimension. Because RoDAN is a member of the DBT method, we require an excellent detector at the front end to provide RoDAN with a set of bounding boxes ($\boldsymbol{\mathcal{B}}_{t}^{i}$, $\boldsymbol{\mathcal{B}}_{t-n}^{j}$), as shown in Fig. 3. However, because object detection belongs to a different computer vision task, we do not provide details on the detector used in this article.

Before elaborating on RoDAN, we first provide a brief description of the original DAN. The current image frame $\boldsymbol{I}_{t}$ and the previous image frame $\boldsymbol{I}_{t-n}$ (n indicates the time stamp) along with the center points of the predetected objects, are input to the front-end convolution (FE-Conv) layer for feature extraction. The FE-Conv layer is a simplified name for the VGG+ extension network in DAN [15]. Then, a specific number of feature maps selected from nine location in the FE-Conv layer are dimensionally reduced to 520. Next, the features corresponding to the center points of each predetected object in each feature map are extracted to form a feature map $\boldsymbol{F}_{t}$ and a feature map $\boldsymbol{F}_{t-n}$. Finally, $\boldsymbol{F}_{t}$ and $\boldsymbol{F}_{t-n}$ are fused in the FF module whose output forms the input of the back-end convolution (BE-Conv) layer, which provides an affinity matrix $\boldsymbol{M}_{t,t-n}$ as the tracking result (note that the BE-Conv layer is a simplified name for the compression network in DAN), and is the core aspect of how DAN works. The DAN pipeline can be seen in the gray boxes shown in Fig. 3. The modules in these gray boxes are retained in RoDAN.

As shown in Fig. 3, the first difference between RoDAN and DAN lies in the input data. The input of RoDAN is an image frame $\boldsymbol{I}_{t}$ (or $\boldsymbol{I}_{t-n}$) along with all the bounding boxes $\boldsymbol{\mathcal{B}}_{t}^{i}$ (or $\boldsymbol{\mathcal{B}}_{t-n}^{j}$) for all the predetected ships in $\boldsymbol{I}_{t}$ (or $\boldsymbol{I}_{t-n}$). Other differences lie in the three dimensions we propose. First, as shown next to the black dashed arrows in Fig. 3, we insert the ASPP into the intervals between the FE-Conv layers to address the tracking challenges of scale variations in the scale dimension. Next, we input the feature maps extracted from the FE-Conv layer into the proposed JGRM module to further enhance the appearance modeling ability of our tracker in the region dimension. As a result, two feature maps $\boldsymbol{F}_{t} \in \mathbb{R}^{Nm\times 520}$ and $\boldsymbol{F}_{t-n} \in \mathbb{R}^{Nm\times 520}$ are individually formed. After obtaining an affinity matrix $\boldsymbol{M}_{t,t-n} \in \mathbb{R}^{Nm\times Nm}$ from the BE-Conv layer (namely, the preliminary tracking results), we fine-tune the preliminary tracking results by exploiting the MMO module in the motion dimension. This not only makes our tracker more robust but also causes it to rely less on the front-end detector and ameliorates long-term occlusions.

In Section III-A–III-C, we elaborate on the three modules from the scale, region, and motion dimensions.

A. Scale Dimension—ASPP

As mentioned above, scale variation makes ship tracking challenging. In addition, as found in our experiments shown in Fig. 2, row 4, we make the interesting assumption that semantic information can also be a cue to address ship tracking. Inspired by the methods exploited in [61]–[64] for semantic segmentation and with the goal of improving the tracker's scale-agnostic and semantic properties, we adopt ASPP in our method, which fuses multiscale features and causes the tracker to learn more holistic and semantic information for each ship. As shown in Fig. 3, we insert the ASPP module between the FE–Conv layers. Table I lists the overall structure of the FE–Conv layer after integrating ASPP.

As the layer number increases, the size of the feature maps will decrease. The disadvantage of small feature maps is that

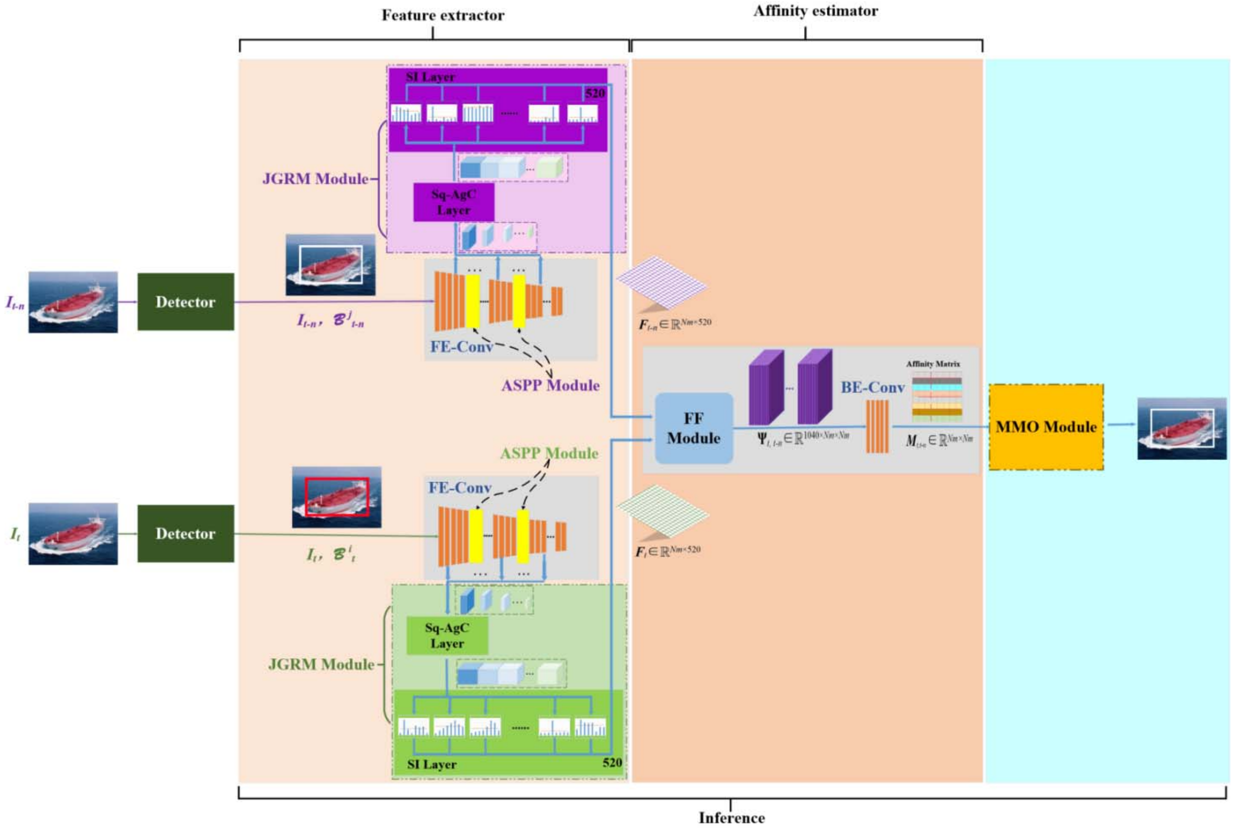


Fig. 3. RoDAN Deployment. As shown, RoDAN deployment involves two steps: a detector and RoDAN. We exploit an off-the-shelf detector that provides a set of bounding boxes ($\mathcal{B}^i_t$, $\mathcal{B}^j_{t-n}$) for each frame. As shown in the figure, RoDAN includes two main streams. The upper stream is used to extract features from the previous frame, while the lower stream is used to extract features for the current frame. Furthermore, we first insert the ASPP module into the intervals between the FE-Conv layers twice (as shown by to the black dashed arrows) to overcome ship scale variations. Next, the features extracted from the FE-Conv are input into the JGRM module to further extract holistic features and overcome the long-tailed distribution property of ships. After obtaining the affinity matrix $\boldsymbol{M}_{t,t-n} \in \mathbb{R}^{Nm \times Nm}$, we input $\boldsymbol{M}_{t,t-n}$ into the MMO module to fine-tune the preliminary tracking results. In the MMO module, all the matched, new, and lost tracks will be rechecked, making our tracker less reliant on the detector and more robust while also ameliorating the problem of long-term occlusions. More details about each module are presented in the next section.

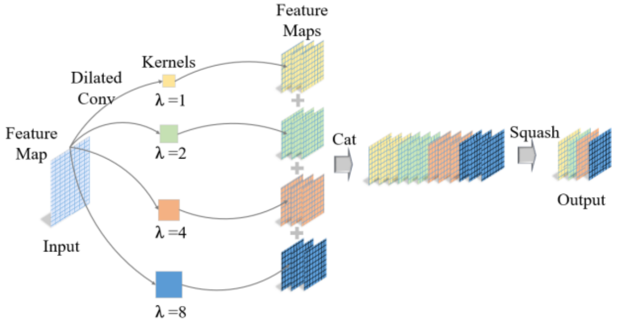


Fig. 4. ASPP Module. Four dilated kernels with different scales are exploited to extract the feature maps; then, the outputs of the dilated convolutions (dilated Conv) are concatenated; finally, a squashing layer combines the feature maps (from 320 × 4 to 256).

they tend to lose information about small objects; likewise, large feature maps lack advanced semantic information. With the goal of extracting holistic features, we want to insert the ASPP modules into different places as far away from each other as possible in the FE–Conv layer. Therefore, we inserted ASPP modules into two locations in the F–EConv layer: layer 16 and layer 56, as shown in bold font in Table I.

Fig. 4 depicts the detailed structure of the ASPP module. First, we exploit four different dilated kernels that are sized with the dilated rate λ to perform a dilated convolution [61]. Then, we obtain feature maps of different scales, as shown in Fig. 4, which shows these different feature maps in yellow, green, orange, and blue. Next, we concatenate (Cat) all these feature maps. Finally, we add a squashing layer (consisting of convolution, rectified linear unit (ReLU), and batch normalization) at the back end to combine the feature maps and alleviate the computing burden.

B. Region-JGRM

We need to strengthen the feature expression ability of DAN, which simply exploits center feature points to express each object. This approach is inadequate for expressing ships, because the long-tailed distribution property causes some center points to be located in the background when tracking ships. We therefore, propose the JGRM module. As shown in Fig. 3, there are two layers in the JGRM: a squashing-aggrandizing concatenation (Sq-AgC) layer and a statistical interpretation (SI) Layer.

Table II lists the detailed structure of the Sq-AgC, which is a compound layer composed of a squashing layer, an aggrandizing layer, and a concatenation layer. We retain the feature selection operation from DAN, which extracts the feature

TABLE I

STRUCTURE OF THE FE-CONV LAYER. NOTE THAT C, B, R, AND M ARE CONVOLUTION, BATCH NORM, RELU, AND MAX POOLING, RESPECTIVELY. FOR EXAMPLE, ASSUME THAT THE INPUT SIZE OF THE FE-CONV IS H × W × 3. WHEN THE LAYER INDEX IS 16, IT CAN BE OBSERVED THAT THE INPUT FEATURE SIZE OF THE ASPP IS 4 × H × W × 256 (1/4 AND 256 ARE THE OUTPUT SIZE AND OUTPUT CHANNEL NUMBER OF LAYER 15, RESPECTIVELY), AND THE OUTPUT CHANNEL NUMBER AND OUTPUT SIZE OF LAYER 16 ARE 256 AND 1/4, RESPECTIVELY

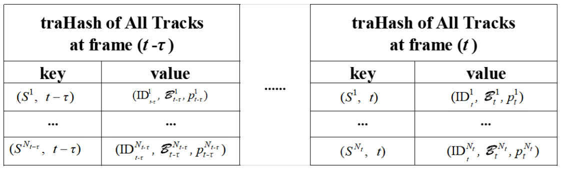
Layer Index	Layer Name	Input Channel	Output Channel	Output Size
0	(C-R)×2	3	64	1
4	M	64	64	1/2
5	(C-R)×2	64	128	1/2
9	M	128	128	1/4
10	(C-R)×3	128	256	1/4
16	**ASPP**	**256**	**256**	1/4
17	(C-R)×1	256	256	1/4
19	M	256	256	1/8
20	(C-R)×3	256	512	1/8
26	M	512	512	1/16
27	(C-R)×3	512	512	1/16
33	M	512	512	1/16
34	(C-R)×2	512	1024	1/16
38	(C-B-R)×6	1024	256	1/64
56	**ASPP**	**256**	**256**	**1/64**
57	(C-B-R)×6	256	256	1/128

TABLE II

DETAILED STRUCTURE OF THE SQ-AGC. THE UNITS IN COLUMN SQUASHING DENOTE THE NUMBER OF CHANNELS OF INPUT (COLUMN B) AND OUTPUT (COLUMN A). THE UNITS IN COLUMN AGGRANDIZING DENOTE THE DOWNSAMPLE STRIDE (FOR EXAMPLE, FOUR IN AGGRANDIZING/B DENOTES THE FEATURE MAPS HERE ARE FOUR TIMES SMALLER IN SIZE THAN THE INPUT FRAME). THE UNIT IN COLUMN CONCATENATION DENOTES THE OUTPUT SIZE OF FEATURE MAPS. NOTE THAT B AND A ARE SHORT FOR BEFORE AND AFTER, RESPECTIVELY

Layer Index	Squashing		Aggrandizing		Concatenation
	B	A	B	A	
0	256	60	4	2	$\frac{1}{2}H\,'\ \ \frac{1}{2}W\,'\ \ 520$
1	512	80	8	2	
2	512	100	4	2	
3	512	80	16	2	
4	256	60	32	2	
5	256	50	32	2	
6	256	40	32	2	
7	256	30	64	2	
8	256	20	64	2	

maps from nine location [15] in the FE-Conv layer. After extracting the feature maps from the FE-Conv layer, we first input these feature maps into the squashing layer (column Squashing in Table II) that also includes of convolution, ReLU, and batch normalization layers to reduce the total number of feature maps and reduce the computational burden. Then, the feature maps are squashed from 3072 (the sum of column B in the squashing operation shown in Table II) to 520 (the sum of column A in the squashing operation in Table II). Next, we aggrandize (column Aggrandizing in Table II) all the feature maps into $(H/2) \times (W/2)$, which substantially reduces the time consumption, because this is the only way can we later extract all the features from the 520 feature maps inside of the detection region simultaneously during the feature extraction process. It is noted that H and W here represent the size of the input frame. Finally, we concatenate all these same-sized feature maps to form the input to the subsequent SI layer.



traHash of All Tracks at frame (t -τ)	
key	value
$(S^1,\ t-\tau)$	$(\mathrm{ID}^1_{t-\tau},\ \mathcal{B}^1_{t-\tau}, p^1_{t-\tau})$
...	...
$(S^{N_{t-\tau}},\ t-\tau)$	$(\mathrm{ID}^{N_{t-\tau}}_{t-\tau},\ \mathcal{B}^{N_{t-\tau}}_{t-\tau}, p^{N_{t-\tau}}_{t-\tau})$

......

traHash of All Tracks at frame (t)	
key	value
$(S^1,\ t)$	$(\mathrm{ID}^1_t,\ \mathcal{B}^1_t, p^1_t)$
...	...
$(S^{N_t},\ t)$	$(\mathrm{ID}^{N_t}_t,\ \mathcal{B}^{N_t}_t, p^{N_t}_t)$

Fig. 5. TraHash tables of all tracks. It is noted that we store only the tracking information for the most recent τ frames. Each row in each traHash table represents a single ship and its tracking information. For instance, to obtain historical information of *ship* 1 in frame $t-1$, we target the traHash table stored at frame $t-1$ and use the ship index 1 to obtain its corresponding tracking information.

The SI layer is exploited to strengthen DAN's modeling ability, because the original DAN exploits only the center feature points of predetected objects to form feature vectors. In contrast, our method exploits SI (e.g., by calculating the quartile or average score) to make full use of all the features within the entire predetected region. Finally, given the time consumption and complexity of the implementation and to avoid the random noise that exists in the feature maps, we adopt the average score in our implementation. After the preparation of the Sq-AgC layer, the 520 feature maps are aggrandized to the same size and concatenated. After obtaining a predetected ship region from the detector, our tracker can calculate the average score inside of this predetected region for each feature map in the SI layer simultaneously. Finally, the 520 average values are stitched together to form a 1×520 feature vector that represents this ship. As shown in Fig. 3, $\boldsymbol{F}_t$ and $\boldsymbol{F}_{t-n}$ are the final feature maps, each row of $\boldsymbol{F}_t$ and $\boldsymbol{F}_{t-n}$ represents a feature vector for a single ship formed in the SI layer, where Nm indicates the number of ships.

C. Motion-MMO

The MMO module is exploited to fine-tune the preliminary tracking results of the affinity matrix $\boldsymbol{M}_{t,t-n}$ and make our tracker more robust, rely less on the front-end detector, and ameliorate long-term occlusions that are endemic to ship tracking. After obtaining the affinity matrix $\boldsymbol{M}_{t,t-n}$, our tracker outputs the preliminary tracking results by associating the predetected ships in the current frame with the tracks that exist in the trajectory gallery. It is noted that the trajectory gallery is a container in which all tracks are stored. As shown in Fig. 5, we utilize a specific number of hash tables to store the tracking information for each track. Concretely, each predetected ship occupies one row of a hash table, which stores the corresponding tracking information. We call this type of hash table the traHash table. As shown in Fig. 5, there are two parts in each traHash table: a key and a value. The key of each traHash table is denoted by $(S^i,\ t)$, which indicates *ship* i in frame t. It is noted that index i is provided by the front-end detector. The value in each traHash table is denoted

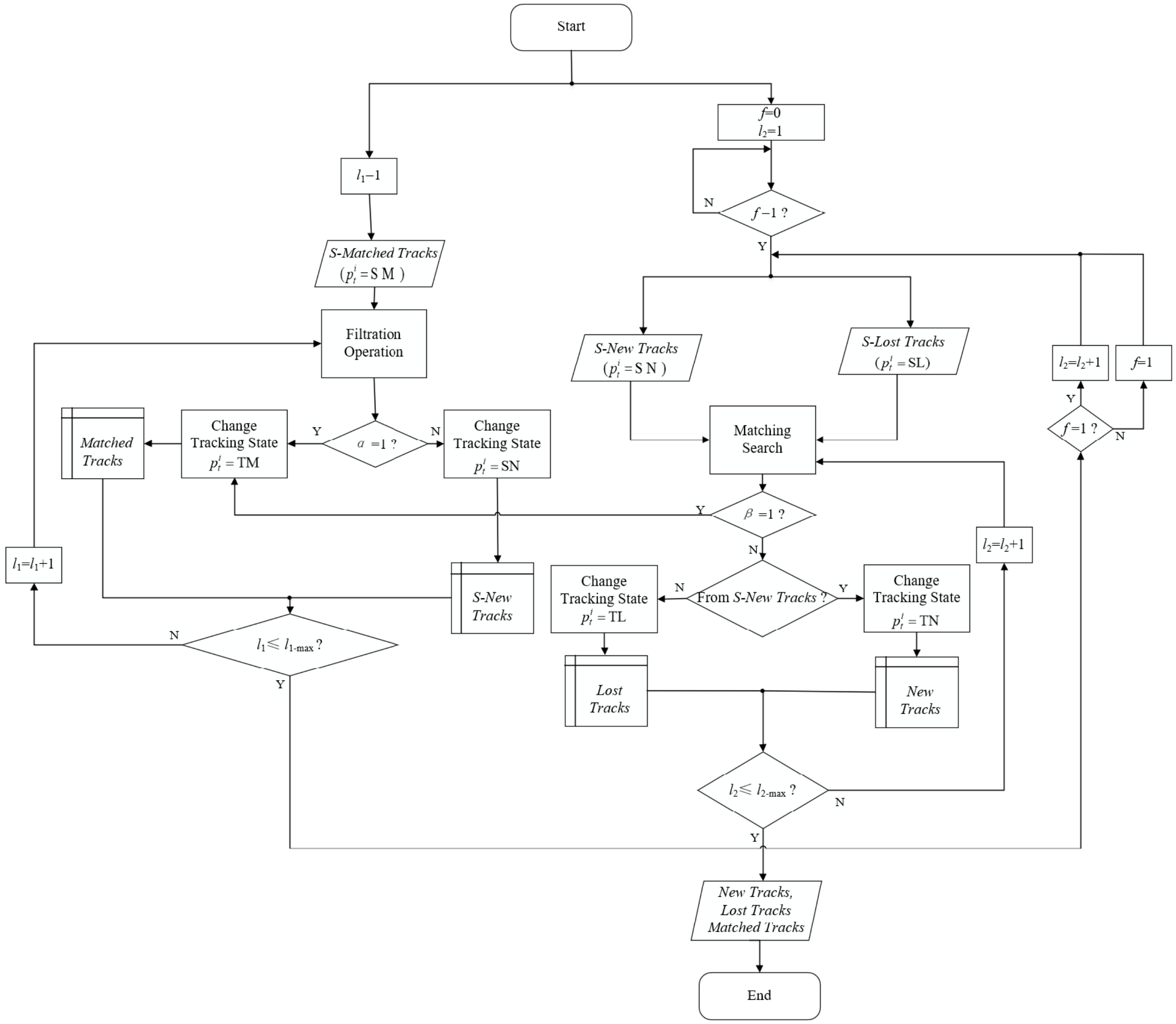


Fig. 6. Flowchart of MMO Module. Note that *S-X Tracks* is short notation for suspicious *X* (matched, new, and lost) *Tracks*. $l_{1-\max}$ is the total number of *S-Matched Tracks,* and $l_{2-\max}$ is the maximum of the total number of *S-New Tracks* and the total number of *S-Lost Tracks*.

by (ID_t^i, $\mathcal{B}_t^i$, p_t^i), which, respectively, indicate the ID of *ship* i at frame t, a bounding box $\mathcal{B}_t^i$, and a tracking state p_t^i. Specifically, if *ship* i at frame t can be matched to one of the previous tracks, we set p_t^i = suspicious matched (SM); if *ship* i newly appears in the current frame, we set p_t^i = suspicious new (SN). For ships that have disappeared in the preliminary tracking results, we save a specific number of data rows in the current traHash table to store corresponding tracking information, the superscript indices of these ships are obtained by sequentially autoincrementing the total number of ships detected in the current frame, and the IDs of these ships are provided by the preliminary tracking results, meaning that the IDs of these ships are consistent with the IDs of those ships that were lost during the preliminary tracking results. At this moment, we set p_t^i = suspicious lost (SL) for all these ships. It is noted that the bounding boxes of the ships whose p_t^i = SL are temporarily ignored. This information storage method makes it easier to perform subsequent information searches. In the remaining part of this section, we elaborate on how to fine-tune all of these tracks by relying on the tracking state p_t^i.

A flowchart of the MMO module is shown in Fig. 6. For each SM track (p_t^i = SM), given that ships usually proceed at low speeds, we perform the filtration operation to ensure correct matches. The filtration operation adheres to the criterion described in the following equation:

$$\alpha(k) = \mathbb{B}\Big[\mathrm{IOU}(\mathcal{B}_t^i, \mathcal{B}_{t-1}^j) \geq \boldsymbol{\Gamma}^1\Big] \tag{1}$$

where k indicates the kth SM track; $\alpha \in \{0, 1\}$ indicates the judging result, e.g., $\alpha = 1$ implies that the kth SM track is correctly matched, and $\alpha = 0$ implies the opposite. $\mathbb{B}$ indicates a binary operation, and IOU is short for intersection over union operation. $\mathcal{B}_t^i$, $\mathcal{B}_{t-1}^j$ indicate the bounding boxes of predetected *ship* i and *ship* j at frame t and frame $t-1$, respectively. It is noted that *ship* i and *ship* j are labeled with an identical ID in the traHash tables; in other words, they belong to an identical track. $\boldsymbol{\Gamma}^1$ indicates the threshold value.

TABLE III

DETAILS OF THE DATASET UTILIZED IN THE EXPERIMENTS. V3 AND V5 INDICATE TWO YOLOV3 AND YOLOV5, RESPECTIVELY

Name	Type	Videos	Length	GT Boxes	Trajectories	Density	YOLO Boxes
SMD	Training	25	12092	120453	241	10.0	36774 (v3)
	Inference	10	4829	40661	83	8.4	15588 (v3)
HSD	Inference	19	24036	113738	98	4.7	45598 (v5)

As shown in (1), we calculate the IOU value simply by taking advantage of the bounding boxes of predetected ships from the most recent two frames, because ships proceed at low speeds. Then, we compare the IOU value with $\mathbf{\Gamma}^1$. When the IOU value is greater than or equal to $\mathbf{\Gamma}^1$, we set $\alpha = 1$, which implies that *ship i* at frame t is correctly tracked, and mark it as a truly matched (TM) ship. At this point, p_t^i is reset to TM; otherwise, p_t^i is reset to SN, and a new ID is assigned to ID_t^i, meaning that this ship is a SN track.

For each SL ($p_t^i = \mathrm{SL}$) track and SN track ($p_t^i = \mathrm{SN}$), we conduct a matching search operation to further enhance the tracking accuracy. Specifically, we walk through each SN track and calculate its similarity with each SL track using (2) with a threshold of $\mathbf{\Gamma}^2$. Then, we find the optimal matched pair and obtain its corresponding similarity value $\beta(m, n) \in \{0.1\}$. When $\beta(m, n) = 1$, *ship i* of the current SN track at frame t and *ship j* of the current SL track at frame t are assumed to be a match; thus, *ship i* and ship j are marked as TM ships, which means, resetting their p_t^i values to TM and assigning an identical ID of *ship j* to ID_t^i in the current traHash table. This acts to extend the track of *ship j* and simultaneously moves *ship j* out of the current traHash table. When $\beta(m, n) = 0$, *ship i* of the current SN track at frame t and *ship j* of the current SL track at frame t are two different ships. In this case, we just reset the tracking state of *ship i* in the current frame t to truly new (TN) in the current traHash table, and no operations for the current SL will be executed

$$\beta(m,n) = \mathbb{B}\Big[\mathrm{IOU}(\mathcal{B}_t^i, \mathcal{B}_{t-1}^j) \geq \mathbf{\Gamma}^2\Big]. \quad (2)$$

It is noted that we utilize the $\mathcal{B}_{t-1}^i$ of the current SL track rather than the $\mathcal{B}_t^i$, because $\mathcal{B}_t^i$ was temporarily vacated as previously discussed.

The IOU exploited in (1) and (2) is given in the following equation:

$$\mathrm{IOU} = \frac{\mathcal{B}_{t_1}^i \cap \mathcal{B}_{t_2}^j}{\mathcal{B}_{t_1}^i \cup \mathcal{B}_{t_2}^j}. \quad (3)$$

At this point, one last problem is how to address the remaining SL ($p_t^i = \mathrm{SL}$) tracks, since these tracks remain unmatched to any predetected ships in the current frame. For this problem, we consider that ship disappearance is a gradual process compared to automobile tracking and pedestrian tracking, because ships usually move at low speeds. This motivates us to perform a stretch processing for these lost tracks. More precisely, we use the most recent bounding box of each lost track, enlarge it slightly to obtain a stretched bounding box and then assign the stretched bounding box to the respective lost track. For the traHash table, we simply replace the previously vacated bounding boxes with the corresponding stretched bounding boxes and reset the tracking state to truly lost (TL) for these TL tracks. During implementation, we also set a threshold $\mathbf{\Gamma}^3$ to limit the number of times a stretching operation will be performed consecutively. This means that if the consecutive stretching time is greater than or equal to $\mathbf{\Gamma}^3$, the current lost track is indeed lost and further stretch processing is no longer needed.

IV. EXPERIMENTS

A. Datasets

Few marine datasets exist in the research community because most applications are commercial or military. Our experiments were first conducted on a public dataset named the SMD [65]. To fully demonstrate the effectiveness of our method, we also conducted experiments on a new marine dataset named the HSD. More details on these two datasets appear below:

1) SMD was collected in Singapore waters using Canon 70D cameras. All the images were captured at a resolution of 1080 × 1920. The SMD dataset is separated into three portions: ship segmentation, ship detection, and ship tracking, making it suitable for three different computer vision tasks. We selected 35 videos annotated by volunteers for our experiments. To verify the effectiveness of our method, we utilized 25 videos for training and 10 videos for testing, as shown in the Type column in Table III. Our training and testing sequences include scenarios with ample types of ships, scale variations, sea fog, dusk illumination, and large waves.
2) HSD is a new dataset suitable for ship tracking. There are 19 videos in HSD, of which 16 were captured by members of our workgroup in Chinese waters, and the other three were taken from maritime detection,classification and tracking (MarDCT) [66]. All the videos are annotated by our workgroup. In our experiments, we exploit all these videos during testing. All the image frames were resized to a resolution of 704 × 450. Compared to SMD, HSD contains more difficult tracking scenarios which occur frequently when tracking ships, including large-scale variations, heavy sea fog, ships far offshore and tiny ships, long-term occlusions, and camera jitter. Therefore, experiments conducted on HSD are more challenging. Quantitative information on these two datasets is presented in Table III.

Each column in Table III shows the total number of videos utilized for ship tracking, the total number of frames in all videos, the total number of ground truth boxes in all frames, the total number of ground truth trajectories, the average number of ships per frame, and the total number of boxes detected using you only look once (YOLO). It is noted that we exploited YOLOv3 [67] for SMD and YOLOv5[1] for HSD, because the scenarios in HSD are more challenging. The weights for YOLOv3 and YOLOv5 are pretrained on COCO[2] without any fine-tuning in our experiments, because a ship class already exists in COCO.

B. Evaluation Metrics

CLEAR MOT [68] and multi-target (MT)/multi-camera (MC) [69] are two evaluation metrics exploited in DAN to quantify the tracking performance. To ensure a fair comparison, we utilized these same evaluation metrics. It is noted that the up-arrow symbols (↑) indicate that larger values are better, while the down-arrow symbols (↓) indicate that smaller values are better.

IDF1 (↑): Identification of the F1 value (harmonic mean value of detection precision and Recall).

Recall (↑): Percentage of detected objects compared to the ground truth objects.

ML (↓): The tracked trajectories that cover less than 20% of the ground truth trajectories during their lifespans.

FP (↓): False positives.

FN (↓): False negatives.

IDS (↓): Identification switch.

FM (↓): Trajectory FM.

MOTA (↑): MOT accuracy combining IDS, FP, and FN.

MOTP (↑): MOT precision indicating the overlaps between the predicted locations and ground truth locations.

Hz (↑): Number of frames that are processed per second.

In addition to the above evaluation metrics, we introduce two new evaluation metrics named continuity of trajectory (CoT) and synthesized MOT accuracy (SMOTA) that evaluate the trajectory performance and tracking performance, respectively. The definition of CoT is given in the following equation:

$$\mathrm{CoT} = 1 - \left(\frac{\sum_{\upsilon} \mathrm{ML}_{\upsilon}}{\sum_{\upsilon} a_{\upsilon}} + \frac{\sum_{\upsilon} \mathrm{FM}_{\upsilon}}{\sum_{\upsilon} b_{\upsilon}} \right) \tag{4}$$

where υ indicates the video index; ML_{υ} and FM_{υ}, respectively, indicate the value of ML computed in video υ and the value of FM computed in video υ; a_{υ} indicates the total number of ground truth trajectories in video υ; and b_{υ} indicates the total number of ground truth boxes in video υ. The CoT can be understood as derived from two error ratios: $\overline{\mathrm{ML}}$ and $\overline{\mathrm{FM}}$. The $\overline{\mathrm{ML}}$ is given in (5) and represents the ratio of the tracked trajectories that cover less than 20% of the ground truth trajectories during their lifespans. The $\overline{\mathrm{FM}}$ is shown in (6) and represents the ratio of trajectory FMs computed over the total number of ground truth boxes that appear in all frames

$$\overline{\mathrm{ML}} = \frac{\sum_{\upsilon} \mathrm{ML}_{\upsilon}}{\sum_{\upsilon} a_{\upsilon}} \tag{5}$$

$$\overline{\mathrm{FM}} = \frac{\sum_{\upsilon} \mathrm{FM}_{\upsilon}}{\sum_{\upsilon} b_{\upsilon}}. \tag{6}$$

TABLE IV

EXPERIMENT CONFIGURATION

Item	Specification
CPU	AMD Ryzen 5 1600 six core
GPU	Nvidia GeForce GTX 1080
Video Memory	12G
Operating System	Ubuntu 18.04
Deep Learning Framework	PyTorch 1.5.1
CUDA	NVIDIA CUDA 10.2

Summing these two different error ratios yields the total error rate E_r, and $1-E_r$ is the final CoT. The CoT accounts for trajectory errors caused by the tracker. It is similar to MOTA but provides trajectory continuity information.

Based on CoT and other existing metrics and given that a good synthesized metric must be simultaneously equipped with the properties of uniformity, necessity, integrity, and low redundancy, we can intuitively define another new metric, SMOTA, as follows

$$\mathrm{SMOTA} = \frac{\mathrm{IDF1} + \mathrm{Recall} + \mathrm{MOTA} + \mathrm{MOTP} + \mathrm{CoT}}{5}. \tag{7}$$

Clearly, SMOTA has the ability to characterize the synthesized tracking performance, as 1) the five metrics used in SMOTA are all necessary for evaluating tracking performance because each metric reveals different information about the tracking performance, fulfilling the properties of necessity and integrity; 2) the five metric used in SMOTA all range within [0, 1], fulfilling the property of uniformity; and 3) the five metrics used in SMOTA are zero-overlapped, fulfilling the property of low redundancy. Therefore, it is more rational to employ the synthesized metric SMOTA to evaluate multiobject tracking performance.

C. Experimental Configuration

The hardware and software configuration of our experiment is listed in Table IV.

D. Training and Inference Details

Training: As shown in Fig. 3, the components used during the training phase are the feature extractor and the affinity estimator. We maintained consistency with DAN regarding parameter initialization; that is, we initialized the parameters of our method using Xavier [70] and trained our method with stochastic gradient descent (SGD) for 50 iterations. We regularized our method with a weight decay of 5e-4 and set the momentum to 0.9 and the initial learning rate (*lr*) to 0.005. We changed the learning rate every 10 iterations in the same way as DAN ($lr = lr \times 0.1^{\mu}$, $\mu \in \{1, 2, 3, 4\}$). Given the GPU memory limitation, we set the batch size to 3 and resized all the images to 650 × 650 to reduce the computational burden. Moreover, as mentioned above, we inserted ASPP at layers 16 and 56 in FE-Conv and utilized $\lambda = \{1, 2, 4, 8\}$ [62] only to verify the effectiveness of adopting the ASPP module.

[1] https://github.com/ultralytics/yoloV5

[2] https://cocodataset.org/#home

TABLE V

RESULTS OF COMPARISON EXPERIMENTS ON SMD. THE BEST RESULTS ARE REPORTED IN BOLD RED FONT

Tracker	IDF1 (↑,%)	Recall (↑,%)	ML (↓)	FP (↓)	FN (↓)	IDS (↓)	FM (↓)	MOTA (↑,%)	MOTP (↑,%)	Hz (↑)	CoT (↑,%)	SMOTA (↑,%)
SORT [25]	38.9	33.8	46	**804**	24836	200	315	31.1	**62.9**	**1448.9**	43.80	42.10
DeepSORT [18]	49.7	38.7	42	1339	23008	126	477	34.8	59.7	14.3	48.22	46.23
DAN	47	39.8	40	1498	22603	212	920	35.2	59.8	6.1	49.55	46.27
RoDAN	**55.7**	**54.3**	**30**	2977	**17158**	**59**	**122**	**46.2**	55.3	4.5	**63.56**	**55.01**

TABLE VI

RESULTS OF COMPARISON EXPERIMENTS ON HSD. THE BEST RESULTS ARE REPORTED IN BOLD RED FONT

Tracker	IDF1 (↑,%)	Recall (↑,%)	ML (↓)	FP (↓)	FN (↓)	IDS (↓)	FM (↓)	MOTA (↑,%)	MOTP (↑,%)	Hz (↑)	CoT (↑,%)	SMOTA (↑,%)
SORT [25]	43.2	36.1	61	**1374**	72749	256	474	34.6	**65.3**	**1865.7**	26.09	41.06
DeepSORT [18]	45.2	34.4	61	7091	74685	177	761	28.0	50.6	41.4	25.84	36.81
DAN	**50.1**	37.6	61	2152	70978	171	1271	35.6	62.0	5.8	26.59	42.38
RoDAN	49.7	**42.5**	**53**	5008	**65466**	**76**	**382**	**38.0**	60.6	4.6	**35.81**	**45.32**

During inference, we eventually set $\boldsymbol{\Gamma}^1$ to 0.25 for SMD and to 0.01 for HSD because the ships in HSD move faster. We set $\boldsymbol{\Gamma}^2$ to 0.05 and $\boldsymbol{\Gamma}^3$ to 50 after conducting a series of experiments. Furthermore, the parameter n in Fig. 3 that indicates the time stamp is set to 15 due to memory limitations, meaning that we compute the 15 most recent affinity matrices $\{\boldsymbol{M}_{t,t-1}, \boldsymbol{M}_{t,t-2}, \ldots, \boldsymbol{M}_{t,t-15}\}$ to link ships in the current frame to ships that exist in the trajectory gallery.

E. Comparison Experiments

1) Experimental Design: To investigate the comprehensive performance of RoDAN, we conducted comparison experiments on SMD and HSD, where we compared our method RoDAN with the baseline method DAN. In addition, because the most common and effective method used for ship tracking is the Kalman filter [65], we also compared our tracker with two other published state-of-the-art online methods, SORT [25] and DeepSORT [18], both of which exploit the Kalman filter to track objects. Furthermore, DeepSORT is also a NN-MOT method.

2) Experimental Results: Comparison results are listed in Tables V and VI. Table V lists the comparison results conducted on SMD, while Table VI shows the comparison results conducted on HSD. From these two tables, we notice that 1) RoDAN yielded the best results on the synthesized metric SMOTA on both datasets SMD and HSD (55.01% on SMD and 45.32% on HSD), with increases of 8.74% and 2.94% compared to DAN, respectively; 2) RoDAN achieved the best results on 9 out of 12 metrics on SMD and 8 out of 12 metrics on HSD, accounting for 75% and 66.7%, respectively; 3) the IDS metric that we are most concerned about achieved the best results on both datasets under RoDAN; and 4) SORT yielded the best results on FP, MOTP, and Hz on both datasets.

F. Ablation Experiments

1) Experimental Design: To evaluate RoDAN and investigate the robustness of the three added modules in working with scale variations, the long-tailed distribution property, and long-term occlusions, we first conducted a set of ablation experiments using the detections from YOLOv3 and YOLOv5 on SMD and HSD.

To further eliminate the influence of detections when tracking, we conducted another set of ablation experiments using detection ground truths that were all manually annotated.

The trackers conducted in the ablation experiments are DAN_A (DAN with ASPP), DAN_M (DAN with MMO), DAN_AJ (DAN_A with JGRM), DAN_AM (DAN_A with MMO), and our complete ship tracker RoDAN (DAN with ASPP, JGRM, and MMO).

2) Experimental Results:

a) Ablation experiments using detections: Table VII lists the tracking results on SMD, while Table VIII shows the tracking results on HSD. As noted in these two tables, we gradually build on DAN by adding the three modules to investigate the effectiveness of each module. For better observation, we plotted the experimental results as line graphs in Fig. 7. The first column in Fig. 7 represents the tracking results on SMD, and the second column in Fig. 7 represents the tracking results on HSD. Based on these experimental results, 1) we can clearly observe (Fig. 7) that Recall, ML, FN, IDS, FM, MOTA, and CoT all achieve better results with the introduction of each module into the DAN models, while FP and MOTP are exceptions and 2) Tables VII and VIII show that FN, IDS,

TABLE VII

Results of Ablation Experiments Using Detections on SMD. Note That a Is Short for ASPP, J Is Short for JGRM, M Is Short for MMO. DAN_A Indicates DAN With A, DAN_AJ Indicates DAN With a and J, DAN_AM Indicates DAN With a and M, RoDAN Indicates DAN With A, J, and M. The Best Results Are Reported in Bold Red Font

Tracker	IDF1 (↑,%)	Recall (↑,%)	ML (↓)	FP (↓)	FN (↓)	IDS (↓)	FM (↓)	MOTA (↑,%)	MOTP (↑,%)	Hz (↑)	CoT (↑,%)	SMOTA (↑,%)
DAN	50.1	37.6	61	**2152**	70978	171	1271	35.6	**62.0**	**5.8**	26.59	42.38
DAN_A	**51.4**	37.6	61	2159	70985	166	1275	35.6	**62.0**	5.5	26.59	42.64
DAN_AJ	50.4	37.6	61	2183	71009	149	1280	35.5	**62.0**	5.0	26.59	42.41
DAN_AM	49.5	42.6	**53**	5113	65307	75	377	**38.0**	60.6	5.4	35.81	45.30
DAN_M	49.3	**42.7**	**53**	5233	**65244**	**73**	**365**	**38.0**	60.3	5.6	**35.82**	45.23
RoDAN	49.7	42.5	**53**	5008	65466	76	382	**38.0**	60.6	4.6	35.81	**45.32**

TABLE VIII

Results of Ablation Experiments Using Detections on HSD. The Best Results Are Reported in Bold Red Font

Tracker	IDF1 (↑,%)	Recall (↑,%)	ML (↓)	FP (↓)	FN (↓)	IDS (↓)	FM (↓)	MOTA (↑,%)	MOTP (↑,%)	Hz (↑)	CoT (↑,%)	SMOTA (↑,%)
DAN	47	39.8	40	1498	22603	212	920	35.2	59.8	**6.1**	49.55	46.27
DAN_A	48.9	39.8	40	1494	22599	205	923	35.2	**59.9**	5.7	49.54	46.67
DAN_AJ	46.5	39.8	40	1498	22608	190	918	35.3	**59.9**	4.6	49.55	46.21
DAN_AM	54.1	**55.8**	28	5164	**16597**	72	120	41.8	54.3	5.6	**65.97**	54.39
DAN_M	54.2	55.7	28	5122	16622	72	**119**	41.9	54.0	5.6	**65.97**	54.35
RoDAN	**55.7**	54.3	**30**	**2977**	17158	**59**	122	**46.2**	55.3	4.5	63.56	**55.01**

and FM all have a tendency to decline substantially compared with RoDAN and DAN—from 22.603 to 16.597, 212 to 59, and 920 to 120 on SMD (Table V), and from 21.555 to 18.271, 66 to 39, and 797 to 227 on HSD (Table VI), respectively.

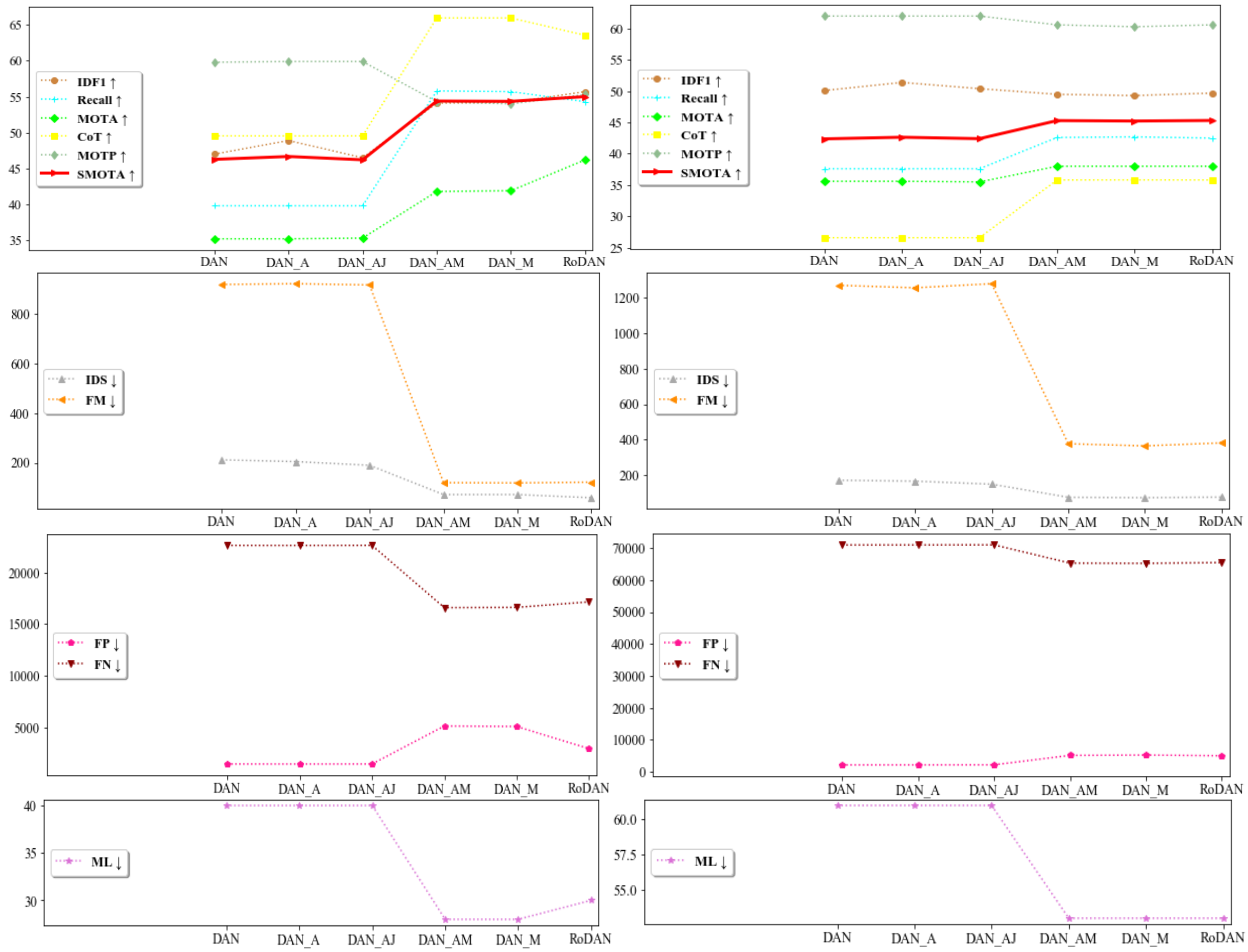
To investigate the continuous tracking capability of the trackers and depict the specific cases of ID switching during experiments, a set of qualitative experiments were carried out as a supplement to the IDS metric as follows.

Fig. 8 represents four tracking examples utilizing DAN and DAN_A (DAN with ASPP). As shown in Fig. 8(a), (b), and (d), under DAN, the IDs of the ships in the initial frames either change IDs or the same ID is switched to a different ship in other frames; however, no switching under DAN_A. In scene Fig. 8(c), ID **13** at frame 284 switches to ID **3** at frame 285 under DAN, but no switching occurs under DAN_A.

Fig. 9 depicts one tracking example of DAN, DAN_A, and DAN_AJ (DAN with ASPP and JGRM). As shown, under DAN, the ID of the ship corresponding to ID 0 at frame 938 switches frequently within the subsequent four frames; however, under DAN_A, the tracking performance improves; nonetheless, ID 0 still switches at frame 984. In contrast, under DAN_AJ, the ID never switches.

Fig. 10 shows three tracking examples comparing DAN and our final tracker RoDAN (DAN with ASPP, JGRM, and MMO). We note that 1) in scene Fig. 10(a), when using DAN for ship tracking, ID **6** at frame 252 switches to **19** at frame 256 and then switches again to **18** at frame 267, but when exploiting RoDAN for ship tracking, ID **6** remains through frame 294, with no switching; 2) in scene Fig. 10(a), when using DAN for ship tracking, ID **0** at frame 252 switches to **29** at frame 275 and then to **38** at frame 294, but when using RoDAN for ship tracking, ID **0** remains through frame 294 with no switching. Similar results occur in scene Fig. 10(b); 3) due to the undesirable performance of the front-end object detector, bounding boxes occur around some people under DAN in scene Fig. 10(a), and the ID frequently switches at the same time; however, this phenomenon entirely disappears under RoDAN; and 4) long-term occlusions resulting from the low sailing speed of ships occur in scene Fig. 10(c), causing DAN to be unable to continue tracking, but our RoDAN tracker can still output the locations of this occluded ship over the next few frames.

Fig. 11 depicts two tracking results of DAN and DAN_A on HSD. Unfortunately, camera jitter occurs at frames 245 and 1281, which causes an IDS under DAN. Interestingly,



(a) (b)

Fig. 7. Quantitative experimental results. As described in the evaluation metrics section, all these metric results should remain consistent with the respective arrow directions, meaning that for metrics with up arrows, larger values are better while for those metrics with down arrows, lower values are better. Therefore, overall, our method yields better tracking results. (a) SMD. (b) HSD.

under DAN_A, the ID never switches. For more experimental results, please refer to HEU-RoDAN-Results.[3]

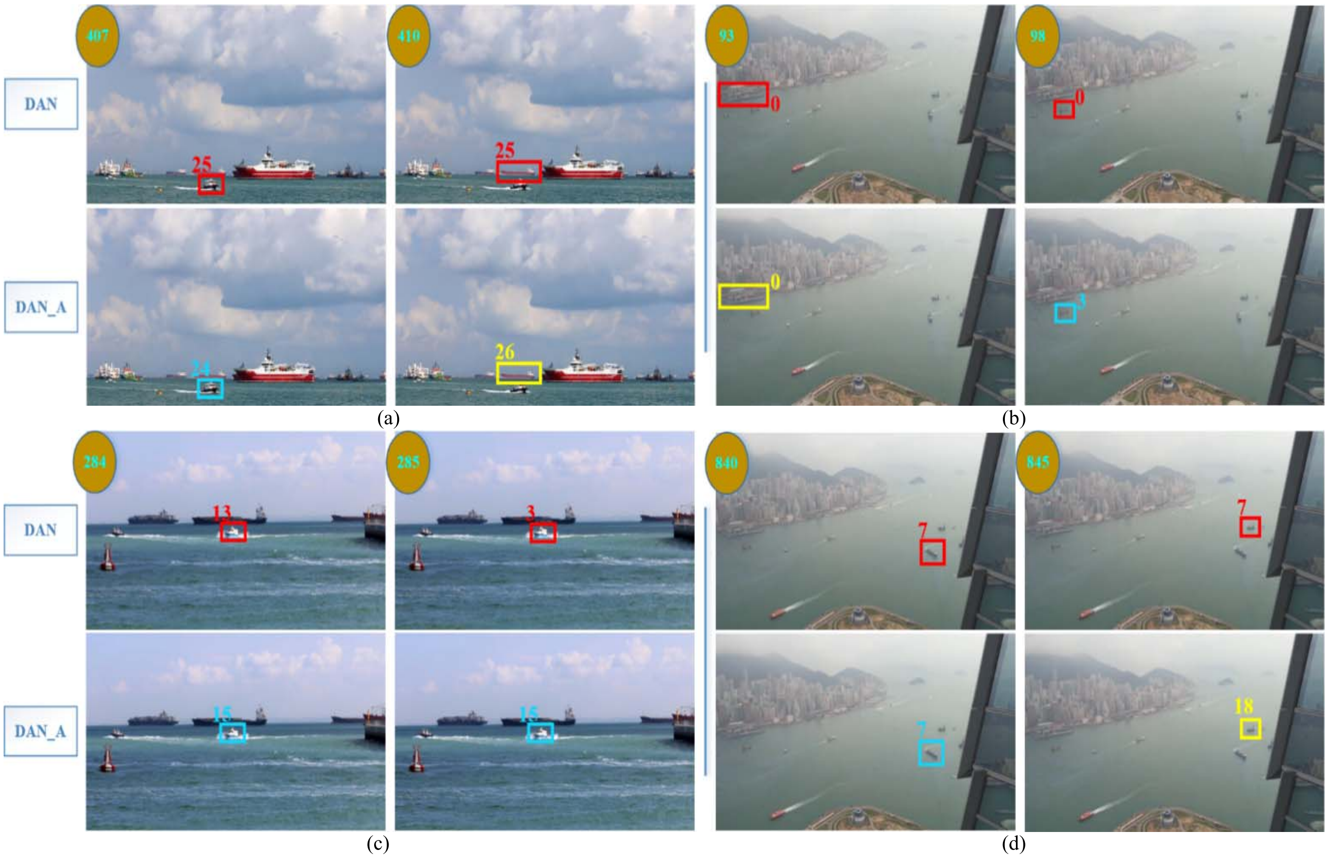
b) Ablation experiments using detection ground truths: In ablation experiments using detections (Tables VII and VIII), we note that the improvements achieved after adding the ASPP and JGRM modules are not as substantial as those achieved after adding the MMO module. This is because the accuracy of the detector can also be influenced by the scale variations and the long-tailed property; in other words, the detections we used have already introduced errors, making the improvement space for ASPP and JGRM less than that for MMO under this situation. However, if the front-end detector can to some extent overcome the scale variations and the long-tailed property, in this case, the ASPP and JGRM we integrated into DAN can fully take effect. This is actually the reason why we organized another set of ablation experiments using detection ground truths.

Table IX lists the tracking results on HSD, while Table X shows the tracking results on SMD. We can see from these two tables that 1) the metric SMOTA that DAN_A yielded is higher than that of DAN_M. Moreover, the metrics FN and FM that DAN_A yielded are all lower than or equal to those of DAN_M, meaning that when using detection ground truths, which is equivalent to using a high-performance detector at the front end when the tracking performance is influenced only by the back-end trackers, in this case, the ASPP module plays a more significant role in ship tracking; 2) after adding the ASPP module to DAN_M, the SMOTA of DAN_AM continues to increase, meaning that the ASPP module has the ability to improve the tracking performance of DAN_M; and 3) under HSD, after adding the JGRM to DAN_A and DAN_AM, the IDS metric continues to decrease, meaning that the JGRM module can further improve the tracking performance.

G. Parameter Determination on $\boldsymbol{\Gamma}^3$

To determine the magnitude of the stretch processing used in the MMO module, we conducted 21 experiments by changing $\boldsymbol{\Gamma}^3$ from 0 to 100 on SMD. The results are listed in Table XI and show that as $\boldsymbol{\Gamma}^3$ increases, the overall SMOTA scores tend

[3] https://github.com/EddieEduardo/HEU-RoDAN-Results



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

Fig. 8. Tracking examples utilizing DAN_A and DAN on SMD [(a) and (c)] and HSD [(b) and (d)]. As shown, due to scale variations, the tracking results of DAN, ID **25** at frame 407 in scene (a) are assigned to another, larger ship in frame 410; however, under DAN_A, no switching occurs (these two ships are consistently regarded as two different ships). Similar tracking results occur in scene (b). In scenes (c) and (d), due to the small ship sizes, ID **13** of DAN at frame 284 in scene (c) switches to **3** at frame 285, but no switching occurs under DAN_A, and similar tracking results occur in scene (d).

TABLE IX
RESULTS OF ABLATION EXPERIMENTS USING DETECTION GROUND TRUTHS ON HSD

Tracker	**IDF1** (↑,%)	**Recall** (↑,%)	**ML** (↓)	**FP** (↓)	**FN** (↓)	**IDS** (↓)	**FM** (↓)	**MOTA** (↑,%)	**MOTP** (↑,%)	**CoT** (↑,%)	**SMOTA** (↑,%)
DAN	91.1	98.8	0	0	1346	431	312	98.4	97.6	99.73	97.13
DAN_M	93.3	99.2	0	10	872	253	183	99.0	97.6	99.84	97.79
DAN_A	99.0	100	0	0	0	15	3	100	97.6	100	99.32
DAN_AJ	99.3	100	0	0	0	13	3	100	97.6	100	99.38
DAN_AM	99.8	100	0	16	0	3	3	100	97.6	100	99.48
RoDAN	99.8	100	0	16	0	1	3	100	97.6	100	99.48

to increase, while the overall results of FN, IDS, and FM tend to decrease.

To further observe the experimental results and determine how $\mathbf{\Gamma}^3$ influences the tracking performance, we calculated the rate of change $\rho(X, \mathbf{\Gamma}^3)$ for FP, FN, IDS, FM, CoT, and SMOTA at each $\mathbf{\Gamma}^3$ point using the following equation:

$$\rho(X, \mathbf{\Gamma}^3) = \nabla X(\mathbf{\Gamma}^3) \approx \frac{\Delta X(\mathbf{\Gamma}^3)}{\Delta \mathbf{\Gamma}^3} = \frac{X(\mathbf{\Gamma}^3 + 5) - X(\mathbf{\Gamma}^3)}{5} \quad (8)$$

where X indicates the evaluation metrics, including FP, FN, IDS, FM, CoT, and SMOTA; X is a function on $\mathbf{\Gamma}^3$; and X indicates extracting the derivatives of function X with respect to the variable $\mathbf{\Gamma}^3$, $\mathbf{\Gamma}^3 \in \{0, 5, 10, 15, \ldots, 95\}$. It is noted that for the metrics FP, FN, IDS, and FM, we take the opposite number for each rate of change $\rho(X, \mathbf{\Gamma}^3)$ because lower values are better for all these metrics, which occur in the opposite direction to their rate of change.

To better observe the results of the parameter determination experiments, we nondimensionalized the results of each metric

TABLE X
RESULTS OF ABLATION EXPERIMENTS USING DETECTION GROUND TRUTHS ON SMD

Tracker	**IDF1** **(↑,%)**	**Recall** **(↑,%)**	**ML** **(↓)**	**FP** **(↓)**	**FN** **(↓)**	**IDS** **(↓)**	**FM** **(↓)**	**MOTA** **(↑,%)**	**MOTP** **(↑,%)**	**CoT** **(↑,%)**	**SMOTA** **(↑,%)**
DAN	99.3	100	0	0	0	63	0	99.8	99.7	100	99.76
DAN_M	99.5	100	0	28	0	2	0	99.9	99.7	100	99.82
DAN_A	99.7	100	0	0	0	24	0	99.9	99.9	100	99.90
DAN_AJ	99.8	100	0	0	0	41	0	99.9	99.9	100	99.92
DAN_AM	100	100	0	28	0	0	0	99.9	99.9	100	99.96
RoDAN	100	100	0	28	0	0	0	99.9	99.9	100	99.96

TABLE XI
RESULTS OF PARAMETER DETERMINATION EXPERIMENTS WITH RODAN

$\mathbf{\Gamma}^3$	**IDF1** **(%, ↑)**	**Recall** **(%, ↑)**	**ML** **(↓)**	**FP** **(↓)**	**FN** **(↓)**	**IDS** **(↓)**	**FM** **(↓)**	**MOTA** **(%, ↑)**	**MOTP** **(%, ↑)**	**CoT** **(%, ↑)**	**SMOTA** **(%, ↑)**
0	38.7	40.5	40	1511	22338	532	562	35	59.9	50.43	44.91
5	41.7	43.9	35	1901	21039	305	345	38.1	58.0	56.98	47.74
10	43.3	46.1	34	2146	20216	216	264	39.8	57.5	58.39	49.02
15	44.5	47.8	34	2253	19591	165	224	41.3	57	58.49	49.82
20	45.9	49.3	34	2421	19019	132	192	42.5	56.6	58.56	50.57
25	48.4	50.2	31	2535	18671	105	164	43.2	56.3	62.25	52.07
30	52.6	51.3	31	2609	18285	83	145	44.1	56.3	62.29	53.32
35	53	52.2	31	2703	17929	74	135	44.8	55.7	62.32	53.60
40	53.5	53	30	2834	17645	68	130	45.2	55.4	63.54	54.13
45	54.6	53.8	30	2904	17322	65	126	45.9	55.2	63.55	54.61
50	55.7	54.3	30	2977	17158	59	122	46.2	55.3	63.56	55.01
55	58.3	54.8	29	3092	16944	49	112	46.5	55.3	64.78	55.94
60	59.8	55.3	28	3195	16780	44	107	46.7	55.6	66.00	56.68
65	60.3	55.9	26	3288	16536	43	109	47.1	55.5	68.41	57.44
70	61.1	56.6	26	3379	16286	41	107	47.5	55.2	68.41	57.76
75	61.9	57.1	26	3454	16105	38	104	47.8	55.1	68.42	58.06
80	61.8	57.7	26	3523	15856	38	104	48.3	54.6	68.42	58.16
85	62.8	58.2	25	3603	15672	36	104	48.5	54.5	69.62	58.72
90	63.5	58.7	25	3642	15497	33	101	48.9	54.4	69.63	59.03
95	64.6	59.1	25	3717	15336	30	100	49.1	54.3	69.63	59.35
100	65.8	59.6	25	3694	15178	28	100	49.6	53.7	69.63	59.67

utilizing the following equation:

$$\rho'(X,\mathbf{\Gamma}^3) = \frac{\rho(X,\mathbf{\Gamma}^3)}{\sqrt{\sum_X \rho^2(X,\mathbf{\Gamma}^3)}}. \tag{9}$$

The nondimensionalized results are listed in Table XII. Additionally, we added another metric $\bar{\rho}'$ in Table XII, which simply computes the average value over each row. We plotted the results of $\bar{\rho}'$ from Table XII as a line graph and performed polynomial curve fitting, as shown in Fig. 12. We note the following results: 1) in Table XII, the absolute value of the nondimensionalized rate of change $|\rho'(X, \Gamma^3)|$ for each metric all decline overall; and 2) in Fig. 12, the values of $\bar{\rho}'$ show a tendency to initially decline sharply and then settle into a more gradual decline after 40 (the region filled with light green in Fig. 12). As shown in Table XII, $\rho'(\text{FM}, 60) = -0.008 < 0$, which indicates the first unexpected value (the expected value should be greater than or equal to 0), and after 75, $\rho'(\text{IDS}, 75) = \rho'(\text{FM}, 75) = \rho'(\text{CoT}, 75) = 0$. These data indicate that the values of these three metrics change just slightly at this point but tend to remain steady. These results actually imply that the preferred interval of $\mathbf{\Gamma}^3$ should be (40, 70) when considering balancing FP from increasing with the synthesized tracking performance. We therefore, set $\mathbf{\Gamma}^3 = 50$ in our experiments.

V. DISCUSSION

In comparison experiments (Tables V and VI), we note that SORT and DeepSORT did not achieve promising results compared to RoDAN. Especially for the SMOTA, MOTA, FM, and IDS metrics, RoDAN yielded far better performance. Actually, in marine scenes, the IDS, SMOTA, MOTA, and FM metrics are more important: with the goal of satisfying the demands of upper-level tasks, such as autonomous ship driving, much more attention is paid to the continuous tracking capability in MST; in other words, the IDS metric is most focused on in MST, especially in observation; to investigate the synthesized performance of the trackers, including the accuracy of the tracking and the accuracy of the detection, we must

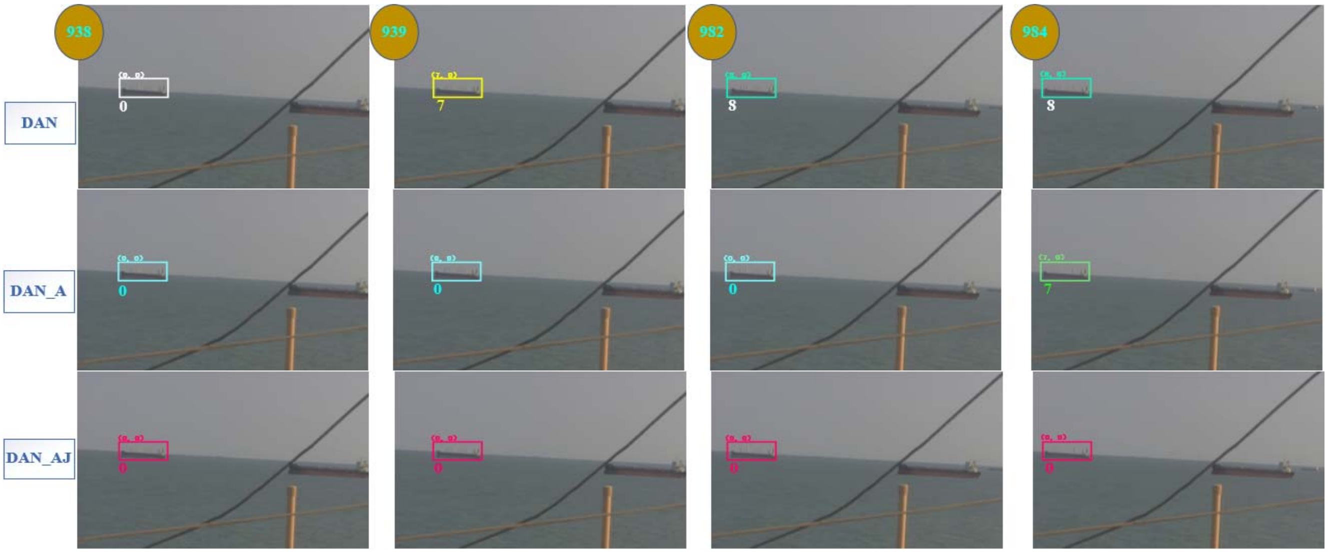
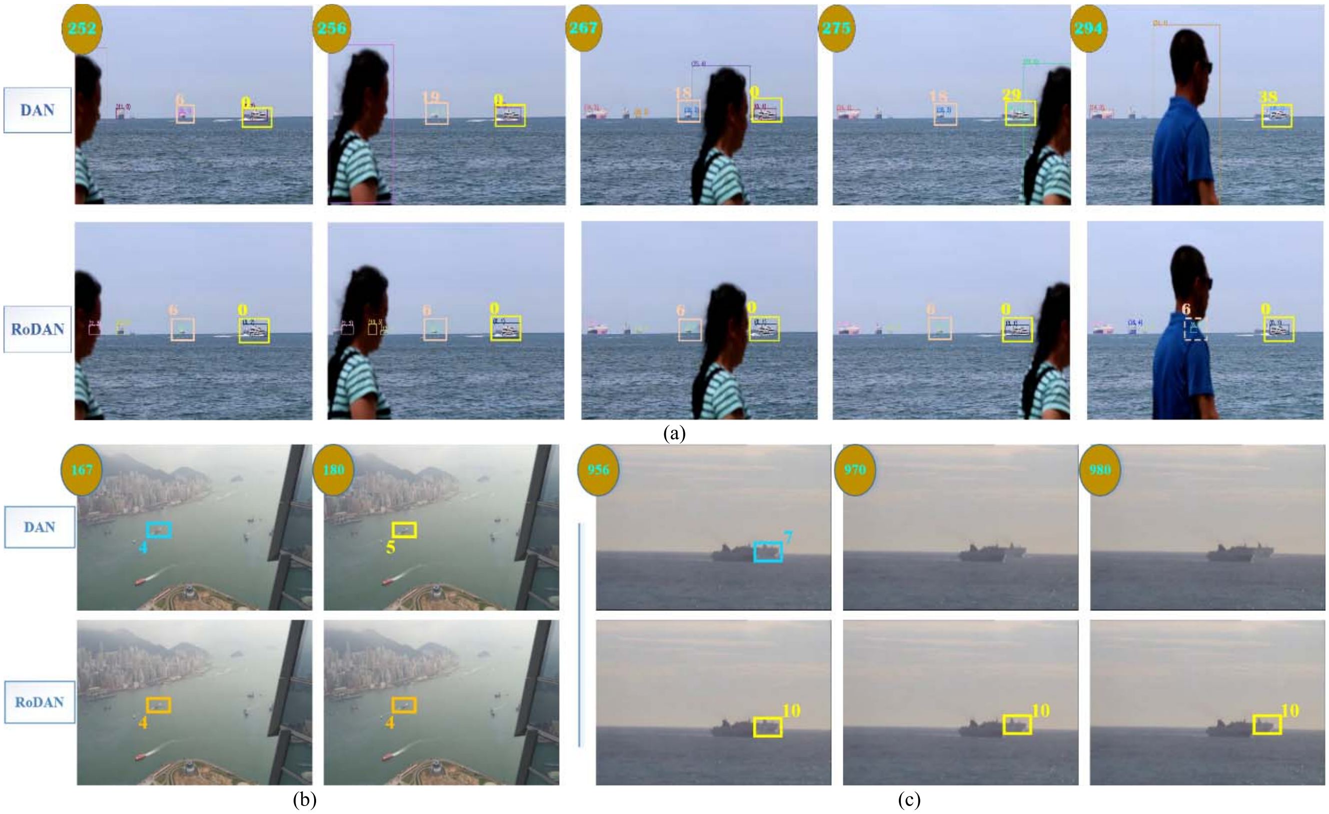


Fig. 9. Tracking examples utilizing DAN, DAN_A and DAN_AJ on HSD. As shown, due to the long-tailed distribution property of ships, under DAN, ID **0** at frame 938 frequently switches over the next four frames; however, under DAN_A, the tracking performance is improved, although an ID switch still occurs at frame 984. In contrast, under DAN_AJ, this ID switch never occurs.



(a)

(b)

(c)

Fig. 10. Tracking examples utilizing DAN and RoDAN on SMD [scene (a)] and HSD [scene (b) and scene (c)]. We captured five identical spots in time from SMD in scene (a). As can be observed, ID **6** of DAN switches frequently within these five frames, but no switching occurs under RoDAN. The same tracking results occur for ID **0** in scene (a) and the ships in scene (b). In scene (c), long-term occlusion occurs due to the low sailing speed; in this case, DAN is unable to continue tracking the occluded ship, but RoDAN can successfully track the location of the occluded ship over the subsequent frames. More discussion regarding the experimental results is provided in Section V.

pay further attention to the SMOTA metric; to investigate only the accuracy of the tracking, we must further pay attention to the MOTA and FM metrics. Based on these reasons, RoDAN is better than SORT and DeepSORT for MST. The reason for SORT is because SORT merely utilizes the Kalman filter and Hungarian matching to perform the tracking, neglecting the appearance information of each object. Based on SORT, DeepSORT integrates the appearance information using a network, but the feature extraction ability of the network that DeepSORT deploys is limited, so the tracking performance

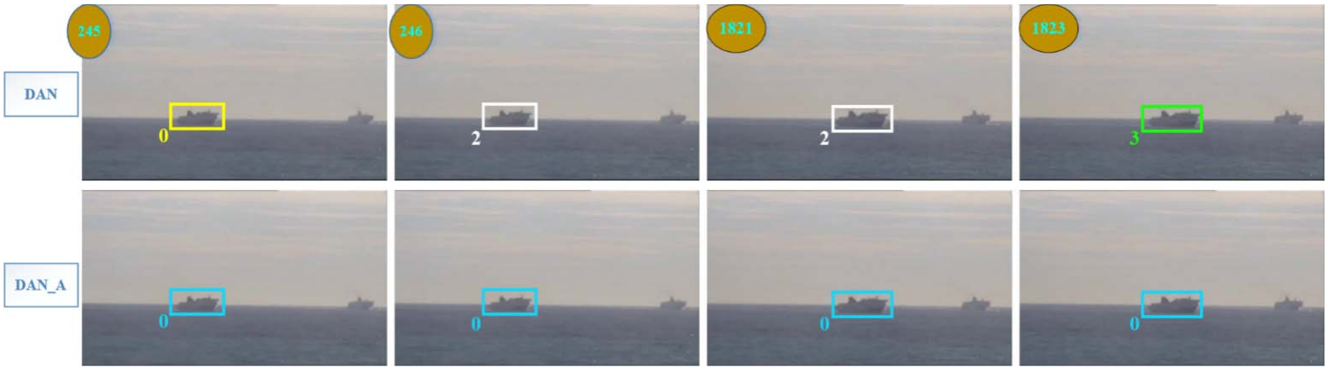


Fig. 11. Tracking examples utilizing DAN and DAN_A on HSD. Camera jitter occurs at frames 245 and 1821. Under DAN, ID **0** switches to another ID at both frames when camera jitter occurs; however, under DAN_A, no switching occurs.

TABLE XII
NONDIMENSIONALIZED RATE OF CHANGE WITH RODAN

$\boldsymbol{\Gamma}^3$	r ¢(FP, $\boldsymbol{\Gamma}^3$)	r ¢(FN, $\boldsymbol{\Gamma}^3$)	r ¢(IDS, $\boldsymbol{\Gamma}^3$)	r ¢(FM, $\boldsymbol{\Gamma}^3$)	r ¢(CoT, $\boldsymbol{\Gamma}^3$)	r ¢(SMOTA, $\boldsymbol{\Gamma}^3$)	$\bar{r}$ ¢
0	-0.652	0.651	0.893	0.903	0.791	0.675	0.544
5	-0.410	0.412	0.350	0.337	0.169	0.305	0.194
10	-0.179	0.313	0.201	0.166	0.012	0.191	0.117
15	-0.281	0.287	0.130	0.133	0.009	0.180	0.076
20	-0.191	0.174	0.106	0.117	0.444	0.357	0.168
25	-0.124	0.193	0.087	0.079	0.006	0.298	0.090
30	-0.157	0.178	0.035	0.042	0.003	0.068	0.028
35	-0.219	0.142	0.024	0.021	0.147	0.125	0.040
40	-0.117	0.162	0.012	0.017	0.001	0.115	0.032
45	-0.122	0.082	0.024	0.017	0.001	0.096	0.016
50	-0.192	0.107	0.039	0.042	0.148	0.221	0.061
55	-0.172	0.082	0.020	0.021	0.147	0.177	0.046
60	-0.156	0.122	0.004	-0.008	0.290	0.181	0.072
65	-0.152	0.125	0.008	0.008	0.001	0.076	0.011
70	-0.125	0.091	0.012	0.012	0.001	0.072	0.010
75	-0.115	0.125	0	0	0	0.024	0.006
80	-0.134	0.092	0.008	0	0.145	0.134	0.041
85	-0.065	0.088	0.012	0.012	0.001	0.072	0.020
90	-0.125	0.081	0.012	0.004	0	0.076	0.008
95	0.038	0.079	0.008	0	0	0.076	0.034

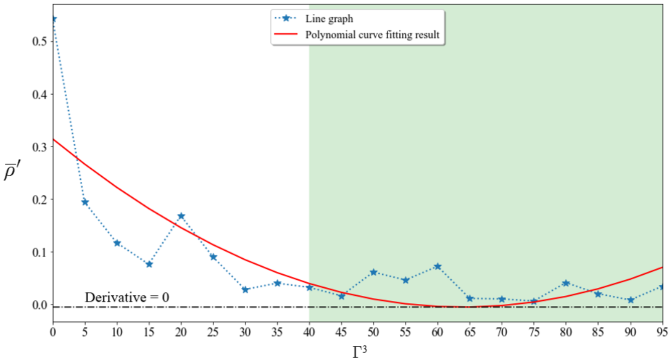


Fig. 12. Visualization of the metric $\bar{\rho}'$ and polynomial curve fitting result.

of these two methods is inferior to that of RoDAN. In our method, with the goal of strengthening the feature extraction ability of the network and overcoming the ID from switching, we proposed RoDAN. To investigate the effects of the three newly proposed modules, we performed two sets of ablation experiments. Regarding the ablation experimental results, we expand on the discussion as follows.

Adopting SMOTA as the evaluation criterion, it is clear (Tables VII and VIII) that DAN_A shows better results than does DAN. This is because DAN_A concentrates more on the ships' multiscale features, and it fuses the multiscale features to obtain the final tracking results. Fusing multiscale features actually prevents the appearance modeling capability from degrading due to ship scale variations. Moreover, as demonstrated in [61]–[64], using features from different scales allows the model to obtain more holistic and semantic information for expressing objects. This also reveals the validity of fusing multiscale features to prevent the appearance modeling capability from degrading because of ship-scale variations.

Adopting SMOTA as the evaluation criterion, it can also be seen (Tables VII and VIII) that the tracking results of DAN_AJ are slightly worse than those of DAN_A. This is acceptable because to further enhance the appearance modeling capability of DAN_A, DAN_AJ improves DAN by exploiting global detection features. It is noted that the feature maps in our method decrease in size as the network becomes deeper. Therefore, it is necessary to interpolate the feature maps back to the same sizes as the regions of predetected ships because only in this way can we employ the detection regions to guide the SI and obtain more accurate results. However, performing interpolation inevitably introduces errors. Of course, increasing the number of training iterations can be a sound solution to compensate. Nonetheless, one subtle aspect that should not be neglected is shown in Tables VII and VIII: the IDS scores of DAN_AJ are smaller than those of DAN_A and DAN in both tables. This actually indicates that DAN_AJ is better able to express ship features and is a more robust tracker.

Taking SMOTA as the evaluation criterion, it can also be observed (Tables VII and VIII) that RoDAN performs the best overall on both datasets. This outcome is inevitable because the MMO module adds the stretch processing to reduce the number of lost tracks as previously described. One prerequisite that should not be neglected is that the affinity matrix $\boldsymbol{M}_{t,t-n}$ should be as accurate as possible. In other words, the tracker should be as robust as possible; otherwise, when the ID of a ship switches frequently, executing the stretch processing for that type of ship will be a poor choice because the incorrect locations will be retained for a few frames, which will drastically degrade the tracking accuracy. Synthesizing the aforementioned discussion, it is easy to grasp why RoDAN performs best.

A close inspection of each evaluation metric shows that (Tables VII and VIII) the experimental results of Recall, ML, FN, IDS, FM, MOTA, and CoT of our methods are all improved, especially FN, IDS, and FM, which gain relative decreases of 24.1%, 72.2%, 86.7% (SMD) and 7.8%, 55.6%, 69.9% (HSD) when comparing RoDAN with DAN, respectively. These results adequately reveal the robustness of RoDAN; that is, RoDAN is able to track multiple ships stably and without interruptions, which is vitally important in MST. The combination of ASPP and JGRM help better model the appearance of each ship, which contributes to obtaining a more accurate affinity matrix $\boldsymbol{M}_{t,t-n}$. A more accurate affinity matrix $\boldsymbol{M}_{t,t-n,}$ as previously described, can impact or even determine the result of the MMO module. The results in Table VII also quantitatively validate this assertion; that is, after adding the ASPP into DAN_M, the SMOTA metric improves to 54.39% (DAN_AM) from 54.35% (DAN_M), and after further adding the JGRM into DAN_AM, the SMOTA metric continues to increase to 55.01% (RoDAN). Identical results can be observed in Table VIII—that is, the SMOTA metric scores of DAN_M, DAN_AM, and RoDAN yield increase from 45.23% to 45.30%, and finally to 45.32%, respectively. Based on all these factors, after performing the filtration operation and the matching search in the MMO module for each track, all the mismatched and interrupted tracks are resolved, which solves the problems of IDS and FM increasing. As mentioned previously, the IDS metric is most focused in MST, but we note that the improvement of IDS tested on HSD is not as much as that tested on SMD. This is because SMD contains ample types of ships; in other words, SMD contains more ships with the long-tailed distribution property, resulting in a more improved IDS value on SMD after further adding the JGRM module. Moreover, initially, we expect MOTP to increase and FP to decrease, but in reality, we found that the opposite occurs, as shown in Tables VII and VIII. Apparently, the increase in FP contributes directly to the decrease in MOTP, because these two metrics are inversely correlated, as explained in [68]. The increase in FP results from the stretch processing exploited in the MMO module. For this problem, to balance the FP from increasing with the synthesized tracking performance, we also performed a series of parameter determination experiments. The experimental results imply that the preferred interval of $\boldsymbol{\Gamma}^3$ should be (40, 70). This gives us a solution to balance the tracking performance while preventing FP from increasing. Actually, the MOTP metric concentrates more on the detection accuracy, especially in MST. We prefer to exploit the DBT methods to satisfy the upper-level demands. In this case, the value of MOTP is also influenced by the front-end detector; as shown in Tables IX and X, if we exploit a high-performance detector (detection ground truths) to track ships, our methods also yield competitive results regarding the MOTP metric. For the Hz metric, RoDAN lags DAN by 1.6 Hz on SMD and 1.2 Hz on HSD because of the different frame sizes and ship densities in these two datasets. Likewise, we also list information regarding the runtime performance of each module in Tables VII and VIII (column Hz) where running time = 1/Hz. We can therefore, find that the running times of ASPP, JGRM, and MMO on SMD are 11.5, 42.0, and 3.1 ms, respectively, and the running times of ASPP, JGRM, and MMO on HSD are 9.4, 18.2, and 3.4 ms, respectively. Clearly, the JGRM module consumes the most time: we implemented this module with no optimization. It is noted that our parameter determination experiments imply that a framewise detection operation is unnecessary, meaning that we could perform ship detection only every few frames rather than every frame. In this case, the tracking time would be substantially accelerated. Of course, slimming the network [71] and removing unnecessary feature extraction layers would also accelerate the processing time for each frame.

From the visualization shown in Fig. 10, we can observe the following: 1) when using DAN to track ships on SMD, ID **6** at frame 252 switches to **19** at frame 256 and then to **18** at frame 267, but under RoDAN, ID **6** remains through frame 294 with no switching. The same tracking results occur for ID **0** in scene Fig. 10(b). The reason RoDAN does not switch is because it benefits from the introduction of the ASPP and JGRM modules. As shown in Fig. 10, a large-scale discrepancy occurs between the lady and the other ships, and the same results occur in Fig. 8, where the ship IDs in Fig. 8 easily switch to other ships with different scales under DAN. In this case, extracting the features from a single scale is apparently inadequate. However, the introduction of ASPP solves these scale problems because ASPP extracts features from different scales and can provide more semantic information. We also introduce the JGRM for each predetected ship region. As shown in Fig. 9, introducing the JGRM further improves our tracker's robustness because the JGRM overcomes the long-tailed distribution property of ships. These characteristics qualitatively certify the effectiveness of the ASPP and JGRM and 2) no bounding boxes for the nonship classes exist in any frames, as shown in the second row of Fig. 10, which benefits from the introduction of the MMO module. Our tracker is specifically designed for MST; however, some people do appear, causing interference that leads to some bounding boxes surrounding people (obtained from the detector) being unstable. These unstable bounding boxes actually reveal that the filtration operation in the MMO module is taking effect, which further reveals that our tracker does not rely too heavily on the front-end detector. Moreover, when ships are occluded, no bounding boxes will be provided by the detector, but the MMO module can still output their locations, as shown in the second row, Frame 294 in Fig. 10, because of the stretch processing. Similar tracking results can be observed in Fig. 10(c), where the ships undergo long-term occlusions. These results reveal that our tracker can ameliorate long-term occlusions. Interestingly, as shown in Fig. 11, when camera jitter occurs, our tracker can still stably track the ships. Under these conditions, the ships are all blurred in the images, and these blurred ships occur over the subsequent few frames as well. At this point, a more powerful feature extraction network (such as RoDAN) can be more effective. The abovementioned results show that our tracker is both more robust and simultaneously less reliant on the detector.

In Tables IX and X, after eliminating the influence of the front-end detector, we note that, if a high-performance detector (detection ground truths) is used in the front end, the ASPP module shows a more significant role in ship tracking. Likewise, in Tables VII and VIII, if a common-performance detector is used in the front end, the MMO module shows a more significant role. Moreover, the results in ablation experiments using detections and using detection ground truths both show that the JGRM module can further prevent the ID from switching. Therefore, to improve the adaption of our tracker to the front-end detector and the final tracking accuracy, we eventually integrate the ASPP, JGRM, and MMO modules in DAN. Furthermore, under SMD, we can also notice that the overall improvements that achieved by SMD are less than those by HSD. This actually also shows that conducting tracking experiments on SMD cannot fully demonstrate the adaption of the trackers to the maritime environment. This is why we created a new dataset; because HSD contains more difficult scenarios that usually occur in marine scenes, conducting experiments on HSD can fully demonstrate the robustness of the trackers to the marine scenarios. Therefore, as shown in Tables V and VI, regardless of the kind of scenario, our tracker RoDAN is able to achieve the best results.

VI. Conclusion

In this article, we present a RoDAN, which fuses tracking information from three dimensions: scale, region, and motion. To obtain more holistic and semantic information to help track ships of different scales, we adopted the ASPP module for the scale dimension. To strengthen the modeling ability of DAN and overcomes the long-tailed distribution property of ships, we proposed the JGRM module for the region dimension. To make our tracker more robust and less reliant on the detector and to ameliorate long-term occlusions, we proposed the MMO module for the motion dimension. The experimental results demonstrate that our method outperforms state-of-the-art methods with respect to the metrics Recall, ML, FN, IDS, FM, MOTA, CoT, and SMOTA, especially IDS and FM. Meanwhile, our method achieves comparable speed and remains simple to implement.

Future work should concentrate on:

1) Optimizing the processing time.
2) Utilizing more discriminative information in the JGRM module, such as segmenting the foreground area for each predetected ship region.
3) Exploiting the Kalman filter in the MMO module to improve the bounding box precision for each ship.

References

[1] D. Frost and J.-R. Tapamo, “Detection and tracking of moving objects in a maritime environment using level set with shape priors,” *EURASIP J. Image Video Process.*, vol. 2013, no. 1, pp. 1–16, Dec. 2013.

[2] Z. L. Szpak and J. R. Tapamo, “Maritime surveillance: Tracking ships inside a dynamic background using a fast level-set,” *Expert Syst. Appl.*, vol. 38, no. 6, pp. 6669–6680, Jun. 2011.

[3] D. Bloisi, L. Iocchi, M. Fiorini, and G. Graziano, “Automatic maritime surveillance with visual target detection,” in *Proc. Int. Defense Homeland Secur. Simul. Workshop (DHSS)*, 2011, pp. 141–145.

[4] M. Kristan, A. Luke, O. Drbohlav, L. He, and Y. Zhang, “The eighth visual object tracking VOT2020 challenge results,” in *Proc. Eur. Conf. Comput. Vis.*, 2020, pp. 547–601.

[5] L. Xiao, M. Xu, and Z. Hu, “Real-time inland CCTV ship tracking,” *Math. Problems Eng.*, vol. 2018, pp. 1–10, Jun. 2018.

[6] M. Yang, T. Yu, and Y. Wu, “Game-theoretic multiple target tracking,” in *Proc. IEEE 11th Int. Conf. Comput. Vis.*, Oct. 2007, pp. 1–8.

[7] L. Zhang and L. van der Maaten, “Preserving structure in model-free tracking,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 36, no. 4, pp. 756–769, Apr. 2014.

[8] W. Hu, X. Li, W. Luo, X. Zhang, S. Maybank, and Z. Zhang, “Single and multiple object tracking using log-Euclidean Riemannian subspace and block-division appearance model,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 34, no. 12, pp. 2420–2440, Dec. 2012.

[9] W. Luo *et al.*, “Multiple object tracking: A literature review,” 2014, *arXiv:1409.7618*. [Online]. Available: http://arxiv.org/abs/1409.7618

[10] J. Peng *et al.*, “Chained-tracker: Chaining paired attentive regression results for end-to-end joint multiple-object detection and tracking,” in *Proc. Eur. Conf. Comput. Vis.* Cham, Switzerland: Springer, 2020, pp. 145–161.

[11] S. Sun, N. Akhtar, X. Song, H. Song, A. Mian, and M. Shah, “Simultaneous detection and tracking with motion modelling for multiple object tracking,” in *Computer Vision—ECCV 2020*. Cham, Switzerland: Springer, 2020, pp. 626–643.

[12] P. Bergmann, T. Meinhardt, and L. Leal-Taixe, “Tracking without bells and whistles,” in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2019, pp. 941–951.

[13] L. Chen, H. Ai, Z. Zhuang, and C. Shang, “Real-time multiple people tracking with deeply learned candidate selection and person re-identification,” in *Proc. IEEE Int. Conf. Multimedia Expo (ICME)*, Jul. 2018, pp. 1–6.

[14] C. Feichtenhofer, A. Pinz, and A. Zisserman, “Detect to track and track to detect,” 2017, *arXiv:1710.03958*. [Online]. Available: http://arxiv.org/abs/1710.03958

[15] S. Sun, N. Akhtar, H. Song, A. S. Mian, and M. Shah, “Deep affinity network for multiple object tracking,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 43, no. 1, pp. 104–119, Jan. 2021.

[16] L. Leal-Taixe, C. Canton-Ferrer, and K. Schindler, “Learning by tracking: Siamese CNN for robust target association,” in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. Workshops (CVPRW)*, Jun. 2016, pp. 418–425.

[17] S. Tang, B. Andres, M. Andriluka, and B. Schiele, “Multi-person tracking by multicut and deep matching,” in *Computer Vision—ECCV 2016 Workshops* (Lecture Notes in Computer Science Including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics), vol. 9914. Cham, Switzerland: Springer, 2016, pp. 100–111.

[18] N. Wojke, A. Bewley, and D. Paulus, “Simple online and realtime tracking with a deep association metric,” in *Proc. IEEE Int. Conf. Image Process. (ICIP)*, Sep. 2017, pp. 3645–3649.

[19] Q. Chu, W. Ouyang, H. Li, X. Wang, B. Liu, and N. Yu, "Online multi-object tracking using CNN-based single object tracker with spatial-temporal attention mechanism," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Oct. 2017, pp. 4846–4855.

[20] A. Sadeghian, A. Alahi, and S. Savarese, "Tracking the untrackable: Learning to track multiple cues with long-term dependencies," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Oct. 2017, pp. 300–311.

[21] C. Kim, F. Li, and J. M. Rehg, "Multi-object tracking with neural gating using bilinear LSTM," *Lect. Notes Comput. Sci. Including Subseries Lect. Notes Artif. Intell. Lect. Notes Bioinform.*, vol. 11212, pp. 208–224, Sep. 2018.

[22] D. Qiao, G. Liu, J. Zhang, Q. Zhang, G. Wu, and F. Dong, "M3C: Multimodel-and-multicue-based tracking by detection of surrounding vessels in maritime environment for USV," *Electronics*, vol. 8, no. 7, p. 723, 2019.

[23] W. Zhang, H. Zhou, S. Sun, Z. Wang, J. Shi, and C. C. Loy, "Robust multi-modality multi-object tracking," *Proc. IEEE Int. Conf. Comput. Vis.*, Oct. 2019, pp. 2365–2374.

[24] D. Bloisi and L. Iocchi, "ARGOS—A video surveillance system for boat traffic monitring in Venice," *Int. J. Pattern Recognit. Artif. Intell.*, vol. 23, no. 7, pp. 1477–1502, 2009.

[25] A. Bewley, Z. Ge, L. Ott, F. Ramos, and B. Upcroft, "Simple online and realtime tracking," in *Proc. IEEE Int. Conf. Image Process. (ICIP)*, Sep. 2016, pp. 3464–3468.

[26] M. H. Assaf, E. M. Petriu, and V. Groza, "Ship track estimation using GPS data and Kalman filter," in *Proc. IEEE Int. Instrum. Meas. Technol. Conf. (I2MTC)*, Houston, TX, USA, May 2018, pp. 1–6, doi: 10.1109/I2MTC.2018.8409579.

[27] S. Hu and B. Yan, "Ship tracking with static electric field based on adaptive progressive update extended Kalman filter," in *Proc. MATEC Web Conf.*, vol. 232, 2018, pp. 1–4.

[28] X. Kang, B. Song, J. Guo, X. Du, and M. Guizani, "A self-selective correlation ship tracking method for smart ocean systems," *Sensors*, vol. 19, no. 4, p. 821, Feb. 2019.

[29] X. Chen, X. Xu, Y. Yang, O. Postolache, Z. Yu, and L. Qi, "Accurate ship tracking in maritime images with kernelized correlation filter and curve fitting," in *Proc. Int. Conf. Sens. Instrum. IoT Era (ISSI)*, Aug. 2019, pp. 1–6.

[30] J. F. Henriques, R. Caseiro, P. Martins, and J. Batista, "High-speed tracking with kernelized correlation filters," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 37, no. 3, pp. 583–596, Mar. 2015.

[31] X. Chen, S. Wang, C. Shi, H. Wu, J. Zhao, and J. Fu, "Robust ship tracking via multi-view learning and sparse representation," *J. Navigat.*, vol. 72, no. 1, pp. 176–192, Jan. 2019.

[32] S. P. van den Broek, P. B. W. Schwering, K. D. Liem, and R. Schleijpen, "Persistent maritime surveillance using multi-sensor feature association and classification," *Proc. SPIE*, vol. 8392, pp. 83920O-1–83920O-11, May 2012.

[33] Y. Liu, L. Yao, W. Xiong, and Z. Zhou, "GF-4 satellite and automatic identification system data fusion for ship tracking," *IEEE Geosci. Remote Sens. Lett.*, vol. 16, no. 2, pp. 281–285, Feb. 2019.

[34] D. Esslinger *et al.*, "Accurate optoacoustic and inertial 3-D pose tracking of moving objects with particle filtering," *IEEE Trans. Instrum. Meas.*, vol. 69, no. 3, pp. 893–906, Mar. 2020.

[35] A. B. Chan and N. Vasconcelos, "Layered dynamic textures," in *Proc. Adv. Neural Inf. Process. Syst.*, 2005, vol. 31, no. 10, pp. 203–210.

[36] A. Mumtaz, W. Zhang, and A. B. Chan, "Joint motion segmentation and background estimation in dynamic scenes," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2014, pp. 368–375.

[37] E. J. Candès, X. Li, Y. Ma, and J. Wright, "Robust principal component analysis?" *J. ACM*, vol. 58, no. 3, pp. 1–37, 2011.

[38] X. Zhou, C. Yang, and W. Yu, "Moving object detection by detecting contiguous outliers in the low-rank representation," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 35, no. 3, pp. 597–610, Mar. 2013.

[39] X. Liu, G. Zhao, J. Yao, and C. Qi, "Background subtraction based on low-rank and structured sparse decomposition," *IEEE Trans. Image Process.*, vol. 24, no. 8, pp. 2502–2514, Aug. 2015.

[40] Q. Tian, K. I.-K. Wang, and Z. Salcic, "A resetting approach for INS and UWB sensor fusion using particle filter for pedestrian tracking," *IEEE Trans. Instrum. Meas.*, vol. 69, no. 8, pp. 5914–5921, Aug. 2020.

[41] W. E. L. Grimson, C. Stauffer, R. Romano, and L. Lee, "Using adaptive tracking to classify and monitor activities in a site," in *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit.*, Jun. 1998, pp. 22–29.

[42] A. Mittal and N. Paragios, "Motion-based background subtraction using adaptive kernel density estimation," in *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit. (CVPR)*, vol. 2, Jun. 2004, p. 7.

[43] V. Ablavsky, "Background models for tracking objects in water," in *Proc. Int. Conf. Image Process.*, vol. 3, Sep. 2003, pp. 125–128.

[44] L. Wixson, "Detecting salient motion by accumulating directionally-consistent flow," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 22, no. 8, pp. 774–780, Aug. 2000.

[45] R. Vidail and Y. Ma, "A unified algebraic approach to 2-D and 3-D motion segmentation," in *Computer Vision—ECCV 2004* (Lecture Notes in Computer Science Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics), vol. 3021. Berlin, Germany: Springer, 2004, pp. 1–15.

[46] D. Cremers and S. Soatto, "Motion competition: A variational approach to piecewise parametric motion segmentation," *Int. J. Comput. Vis.*, vol. 62, no. 3, pp. 249–265, May 2005.

[47] T. Amiaz and N. Kiryati, "Piecewise-smooth dense optical flow via level sets," *Int. J. Comput. Vis.*, vol. 68, no. 2, pp. 111–124, Jun. 2006.

[48] R. Tron and R. Vidal, "A benchmark for the comparison of 3-D motion segmentation algorithms," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2007, pp. 1–8.

[49] Y. Sheikh, O. Javed, and T. Kanade, "Background subtraction for freely moving cameras," in *Proc. IEEE 12th Int. Conf. Comput. Vis.*, Sep. 2009, pp. 1219–1225.

[50] T. Brox and J. Malik, "Object segmentation by long term analysis of point trajectories," in *Computer Vision—ECCV 2010* (Lecture Notes in Computer Science Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics), vol. 6315, no. 5. Berlin, Germany: Springer, pp. 282–295, 2010.

[51] R. Vidal, "Subspace clustering," *IEEE Signal Process. Mag.*, vol. 28, no. 2, pp. 52–68, Mar. 2011.

[52] V. Nair and J. J. Clark, "An unsupervised, online learning framework for moving object detection," in *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit. (CVPR)*, vol. 2, Jun. 2004, pp. 3–10.

[53] H. Grabner, M. Grabner, and H. Bischof, "Real-time tracking via on-line boosting," in *Proc. Brit. Mach. Vis. Conf.*, 2006, pp. 47–56.

[54] H. Grabner, C. Leistner, and H. Bischof, "Semi-supervised on-line boosting for robust tracking," in *Proc. Eur. Conf. Comput. Vis.*, 2008, no. 813399, pp. 234–247.

[55] A. Adam, E. Rivlin, and I. Shimshoni, "Robust fragments-based tracking using the integral histogram," in *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit. (CVPR)*, vol. 1, Jun. 2006, pp. 798–805.

[56] B. Babenko, M.-H. Yang, and S. Belongie, "Robust object tracking with online multiple instance learning," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 33, no. 8, pp. 1619–1632, Aug. 2011.

[57] D. A. Ross, J. Lim, R.-S. Lin, and M.-H. Yang, "Incremental learning for robust visual tracking," *Int. J. Comput. Vis.*, vol. 77, nos. 1–3, pp. 125–141, May 2008.

[58] M. J. Black and A. D. Jepson, "Eigentracking: Robust matching and tracking of articulated objects using a view-based representation," in *Computer Vision—ECCV'96* (Lecture Notes in Computer Science Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics), vol. 1064. Berlin, Germany: Springer, 1996, pp. 329–342.

[59] F. Yin and F. Gunnarsson, "Distributed recursive Gaussian processes for RSS map applied to target tracking," *IEEE J. Sel. Topics Signal Process.*, vol. 11, no. 3, pp. 492–503, Apr. 2017.

[60] S. Chopra, R. Hadsell, and Y. LeCun, "Learning a similarity metric discriminatively, with application to face verification," in *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit. (CVPR)*, vol. 1, Jun. 2005, pp. 539–546.

[61] L. C. Chen, G. Papandreou, I. Kokkinos, K. Murphy, and A. L. Yuille, "Semantic image segmentation with deep convolutional nets and fully connected CRFs," 2014, *arXiv:1412.7062*. [Online]. Available: http://arxiv.org/abs/1412.7062

[62] L.-C. Chen, G. Papandreou, I. Kokkinos, K. Murphy, and A. L. Yuille, "DeepLab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected CRFs," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 40, no. 4, pp. 834–848, Apr. 2018.

[63] L.-C. Chen, G. Papandreou, F. Schroff, and H. Adam, "Rethinking atrous convolution for semantic image segmentation," 2017, *arXiv:1706.05587*. [Online]. Available: http://arxiv.org/abs/1706.05587

[64] L. C. Chen, Y. Zhu, G. Papandreou, F. Schroff, and H. Adam, "Encoder-decoder with atrous separable convolution for semantic image segmentation," in *Lect. Notes Comput. Sci. Including Subseries Lect. Notes Artif. Intell. Lect. Notes Bioinform.*, vol. 11211, pp. 833–851, Aug. 2018.

[65] D. K. Prasad, D. Rajan, L. Rachmawati, E. Rajabally, and C. Quek, "Video processing from electro-optical sensors for object detection and tracking in a maritime environment: A survey," *IEEE Trans. Intell. Transp. Syst.*, vol. 18, no. 8, pp. 1993–2016, Aug. 2017.

[66] E. Gundogdu, B. Solmaz, V. Yücesoy, and A. Koc, "Marvel: A large-scale image dataset for maritime vessels," in *Comput. Vis.—ACCV 2016*. Cham, Switzerland: Springer, 2017, pp. 165–180.

[67] J. Redmon and A. Farhadi, "YOLOv3: An incremental improvement," 2018, *arXiv:1804.02767*. [Online]. Available: http://arxiv.org/abs/1804.02767

[68] K. Bernardin and R. Stiefelhagen, "Evaluating multiple object tracking performance: The CLEAR MOT metrics," *EURASIP J. Image Video Process.*, vol. 2008, pp. 1–10, Dec. 2008.

[69] E. Ristani, F. Solera, R. Zou, R. Cucchiara, and C. Tomasi, "Performance measures and a data set for multi-target, multi-camera tracking," *Computer Vision—ECCV 2016 Workshops* (Lecture Notes in Computer Science Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics), vol. 9914. Cham, Switzerland: Springer, 2016, pp. 17–35.

[70] X. Glorot and Y. Bengio, "Understanding the difficulty of training deep feedforward neural networks," *J. Mach. Learn. Res.*, vol. 9, pp. 249–256, May 2010.

[71] Z. Liu, J. Li, Z. Shen, G. Huang, S. Yan, and C. Zhang, "Learning efficient convolutional networks through network slimming," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Oct. 2017, pp. 2755–2763.

Wen Zhang received the Ph.D. degree from Harbin Engineering University, Harbin, China, in 2010.

She is currently an Associate Professor with the College of Intelligent Systems Science and Engineering, Harbin Engineering University. She has received research funds from the Chinese Natural Science Foundation, Beijing, China, and participated in several intelligent ship projects. Her research interests include intelligent ship systems, object detection and tracking in maritime environments, and ship traffic simulation.

Xujie He received the B.Eng. degree from Shenyang Aerospace University, Shenyang, China, in 2018. He is currently pursuing the M.Eng. degree in automation science and engineering, Harbin Engineering University, Harbin, China.

His research interests include semantic ship segmentation and multiple ship tracking.

Wanyi Li received the M.Eng. degree in computer science from Guizhou University, Guiyang, China, in 2010, and the Ph.D. degree from the Institute of Automation, Chinese Academy of Sciences, Beijing, China, in 2014.

He is currently an Associate Professor with the Institute of Automation, Chinese Academy of Sciences. His research interests include computer vision and machine learning.

Zhi Zhang received the Ph.D. degree from Harbin Engineering University, Harbin, China, in 2007.

He is currently an Associate Professor with the College of Intelligent Systems Science and Engineering, Harbin Engineering University, and a Member of the Leading Goose Team in Heilongjiang province. He has presided more than 20 programs, including the National Natural Science Foundation Program, the High-tech Ship Key Research Program, and the Naval Lateral Scientific Research Program. His research interests include robotics, system simulation, intelligent ship systems, computer vision, and artificial intelligence.

Yongkang Luo received the B.Eng. degree from the South China University of Technology, Guangzhou, China, in 2006, and the Ph.D. degree from the Institute of Automation, Chinese of Sciences, Beijing, China, in 2016.

He is currently with the State Key Laboratory of Management and Control for Complex Systems, Institute of Automation, Chinese Academy of Sciences, Beijing. His current research interests include computer vision, robotics, and machine learning.

Li Su received the Ph.D. degree from Harbin Engineering University, Harbin, China, in 2006.

She is currently an Associate Professor with the College of Intelligent Systems Science and Engineering, Harbin Engineering University. Her research interests include-environment perception and intelligent control, object detection, and multisensor information fusion.

Dr. Su was a recipient of the National Defense Science and Technology Progress Award once, the Science and Technology Progress Award of Heilongjiang Province twice, and the Oceanic Engineering and Technology Award once.

Peng Wang received the B.Eng. degree in electrical engineering and automation from Harbin Engineering University, Harbin, China, in 2004, the M.Eng. degree in automation science and engineering from the Harbin Institute of Technology, Harbin, in 2007, and the Ph.D. degree from the Institute of Automation, Chinese Academy of Sciences, Beijing, China, in 2010.

He is currently a Professor with the State Key Laboratory of Management and Control for Complex Systems, Institute of Automation, Chinese Academy of Sciences. His current research interests include intelligent robots, robotic vision, and image processing and visual attention models.