THE JOURNAL OF NAVIGATION (2019), **72**, 176–192. © The Royal Institute of Navigation 2018
doi:10.1017/S0373463318000504

Robust Ship Tracking via Multi-view Learning and Sparse Representation

Xinqiang Chen[1], Shengzheng Wang[2], Chaojian Shi[2], Huafeng Wu[2], Jiansen Zhao[2] and Junjie Fu[2]

[1] *(Institute of Logistics Science and Engineering, Shanghai Maritime University, Shanghai, 201306, PR China)*
[2] *(Merchant Marine College, Shanghai Maritime University, Shanghai, 201306, PR China)*
(E-mail: chenxinqiang@stu.shmtu.edu.cn)

Conventional visual ship tracking methods employ single and shallow features for the ship tracking task, which may fail when a ship presents a different appearance and shape in maritime surveillance videos. To overcome this difficulty, we propose to employ a multi-view learning algorithm to extract a highly coupled and robust ship descriptor from multiple distinct ship feature sets. First, we explore multiple distinct ship feature sets consisting of a Laplacian-of-Gaussian (LoG) descriptor, a Local Binary Patterns (LBP) descriptor, a Gabor filter, a Histogram of Oriented Gradients (HOG) descriptor and a Canny descriptor, which present geometry structure, texture and contour information, and more. Then, we propose a framework for integrating a multi-view learning algorithm and a sparse representation method to track ships efficiently and effectively. Finally, our framework is evaluated in four typical maritime surveillance scenarios. The experimental results show that the proposed framework outperforms the conventional and typical ship tracking methods.

KEYWORDS

1. Ship tracking. 2. Multi-view learning. 3. Sparse representation. 4. Smart ship.

Submitted: 2 March 2017. Accepted: 19 June 2018. First published online: 13 September 2018.

1. INTRODUCTION. The “smart ship”, which can be defined as a vessel which can automatically collect data, assess its environment and make intelligent sailing decisions, is a future development trend in the shipping industry as it has the potential to lower risks to crew at sea, improve maritime traffic safety (Statheros et al., 2008; Wang, 2010) and significantly reduce the costs to general manufacturing industry. Smart ships will be complicated and novel vessels which can detect their navigation surroundings autonomously and make sailing decisions without a human being’s involvement (Burmeister et al., 2014; Statheros et al., 2008). Thus, it can be seen that the smart ship will remould the future operation modes of the shipping industry. Ship tracking is one of the critical technologies to facilatate the ability of smart ships to make sailing decisions and has attracted the attention of many researchers. Numerous studies have been completed which are relevant to the ship tracking

topic, such as tracking and attacking intrusive ships (Gray et al., 2011; 2012; Han et al., 2014), illegal fishing (Hu et al., 2011), anti-piracy (Szpak and Tapamo, 2011) and so on. We do not list studies of ship tracking in darkness as our work does not cover this aspect.

Previous studies have shown that image processing-based methods can achieve satisfactory results in ship tracking. Ma et al. (2011) tried to find target ships in Vessel Traffic Services (VTS) videos with a Kalman filter algorithm. Considering ships recorded in videos with different viewpoints, Loomans et al. (2013) proposed an efficient object descriptor using a visual feature point tracker to carry out ship tracking tasks robustly. Teng et al. (2013; 2014) and Teng and Liu (2014) conducted several studies to track ships sailing in inland waterways in real time. Some researchers have employed Automatic Identification System (AIS) and Synthetic Aperture Radar (SAR) technologies to track ships (Chaturvedi et al., 2012; Zhao et al., 2014). From the perspective of traditional navigation, the above-mentioned ship tracking methods can sufficiently meet the ship-tracking demands of maritime safety administrations, ship owners, etc. In the smart ship era, crews will be smaller, resulting in challenges for the autonomous ship tracking task.

Firstly, the models used for ship tracking need to be very robust in varied environments, such as lighting variation, extreme weather, different viewpoints, etc. Conventional ship tracking methods may be ineffective in handling such challenges. For instance, SAR-based technologies have a degraded tracking performance under conditions of rain, snow and strong sea clutter. In addition, the disadvantage of AIS is that not all ships have been equipped with AIS, and ships at sea may deactivate their AIS transmitters (for example, warships, ships engaged in illegal activity, etc) (Xiao et al., 2015; Zhang et al., 2015; 2016; 2017a). In such a situation, other techniques are required to help AIS-based methods track ships. Another ship tracking technique is Long Range Identification and Tracking (LRIT). Although the LRIT system provides accurate ship positions, it sends out ship information every six hours at best, far from all ships participate in LRIT, and LRIT data is seldom accessed by merchant ships but by shore authorities such as flag state, port or administration (Chen, 2014). The data interval is too long for ship tracking in the smart ship era. Also, merchant ships may not be able to access LRIT data as the dataset is confidential (Lapinski et al., 2016; Vespe et al., 2015).

Secondly, ship video information (such as ship's imaging width and length) is crucial for a smart ship making sailing decisions. However, little attention has been paid to obtaining visual information in the existing ship tracking models. Thirdly, from the perspective of image processing, ship tracking in the smart ship era requires a higher accuracy compared to that in the traditional navigation era. Hence, efficient computer vision-based models are needed to assist in tackling the above-mentioned ship tracking challenges. We anticipate that a smart ship would employ both visual and non-visual tracking models to obtain a robust ship tracking performance.

Smart ships will require robust visual ship tracking methods to cope with different and challenging tracking situations. Several computer vision and machine learning methods have presented favourable results in the object tracking field (Joshi and Thakore, 2012; Tang et al., 2017; Teng et al., 2013; Yan et al., 2017). Optical flow operators, Kalman trackers, Camshift and Meanshift descriptors are popular object tracking methods, which have shown their potential in ship tracking in smart ship applications (Allen et al., 2004; Bardow et al., 2016; Chauhan and Krishan, 2013; Tripathi et al., 2016). These methods track a target according to a target's single feature. The problem is that we cannot determine a universal single feature which can be applied in different tracking situations. Thus, these models may be inefficient in demanding ship-tracking situations (Tang et al., 2013; 2015; Xu et al., 2013; Zhang et al., 2017b).

Recently, multi-view learning-based methods have been proposed to solve the ineffectiveness of single-feature tracking methods. A bank of distinctive visual features, such as colour, intensity, edge and texture, are integrated into multi-view learning methods to achieve robust object tracking. Many studies have shown excellent tracking performance of multi-view methods (Delamarre and Faugeras, 1999; Hong et al., 2013; 2015; Taj and Cavallaro, 2010). Impressed by the high performance of multi-view learning methods, a novel multi-view learning-based ship tracking framework is proposed in the context of smart ship applications. The proposed framework employs a multi-view learning method to explore intrinsic relationships between different unique ship-contour related features including Laplacian-of-Gaussian (LoG), Local Binary Patterns (LBP), Gabor, Histogram of Oriented Gradients (HOG) and Canny descriptors. A sparse representation method is then applied to represent mutual relations between the distinct features. Traditional powerful tracking methods, including Kalman and Meanshift trackers, have been implemented to verify the proposed methodology's efficiency and accuracy. Combining visual and traditional tracking methods can enhance maritime traffic safety considerably.

Our contributions can be summarised as follows. First, as there is no publicly accessible benchmark for evaluating ship trackers' performance, we collected four typical video datasets for assessing different ship trackers' performance. Second, we analysed bottlenecks existing in the visual ship tracking task. To solve this problem, we introduce a novel visual ship tracking framework based on a multi-view learning model and a sparse representation method. Third, the standard multi-view learning model employs appearance-based feature descriptors to obtain a robust tracking performance. Specifically, colour-related features play an important role in obtaining high tracking performance for conventional multi-view learning models. However, ships sailing in waterways often present similar or identical colours. Also, it is quite common for ships to have similar shapes. It is quite possible that a target ship has a similar appearance to its neighbours in tracking videos. Thus, the standard multi-view-based models, tracking through appearance-based feature sets, cannot deliver a satisfactory ship-tracking performance. However, it is worth noting that different ships possess unique textures and contours. Thus, we propose a novel ship tracking model based on a customised multi-view learning framework, which employs distinctive texture and contour-related features to complete the visual ship-tracking task.

2. METHODOLOGY FOR VISUAL SHIP TRACKING.

2.1. *Multi-view learning and sparse representation-based ship tracker.* A ship tracking model needs to track ships in the Region Of Interest (ROI) with robust performance and low computational complexity. A Lasso regularisation (L_1-norm)-based tracker, abbreviated as L_1 tracker, is a popular object tracker, and it shows advantageous tracking accuracy with a short computation time (Friedman et al., 2008; Mei and Ling, 2011). Hong et al. (2013) pointed out that the L_1 tracker may fail to track objects whose shape varies in tracking image sequences. Thus, a multi-view learning-based method was proposed to improve the L_1 tracker's performance. Building on this work, we propose a novel ship tracking framework based on multi-view learning and a sparse representation method. The flowchart of our proposed ship tracker is shown in Figure 1.

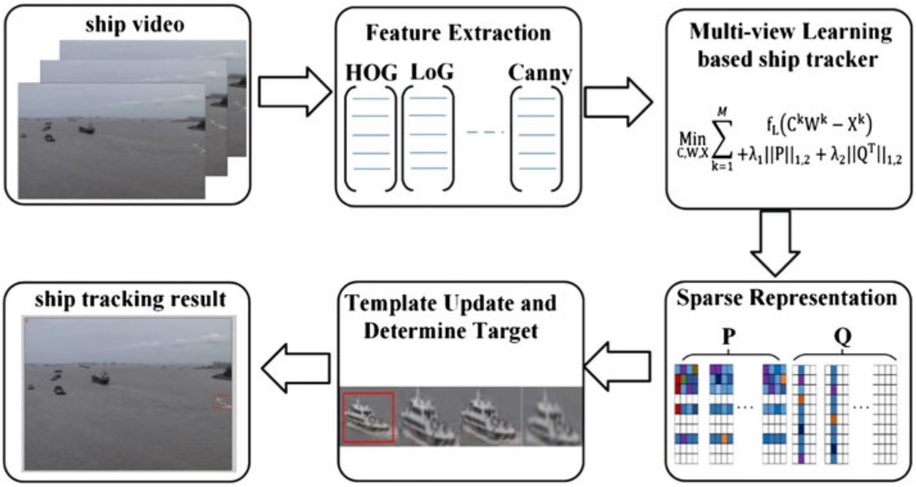

ship video
Feature Extraction
HOG LoG
Canny
Multi-view Learning based ship tracker
$\min_{C,W,X} \sum_{k=1}^{M} f_L(C^kW^k - X^k) + \lambda_1\|P\|_{1,2} + \lambda_2\|Q^T\|_{1,2}$
Sparse Representation
P
Q
Template Update and Determine Target
ship tracking result

Figure 1. Workflow of the proposed ship tracker.

2.1.1. *Feature extraction.* Ships have clear contours and edges compared to water in videos. Hence, texture and structure-based feature descriptors are preferred for extracting ship features in the ROI. HOG and LoG descriptors are two useful texture descriptors which have shown success in many tracking tasks (Dalal and Triggs, 2005; Gunn, 1999; Yang et al., 2012). Details of obtaining a HOG descriptor can be found in Dalal and Triggs (2005) and Suard et al. (2006). The LoG descriptor is a popular blob-detector method (Silvas et al., 2016), which suits well for visual ship tracking tasks. The LBP descriptor possesses features of rotation and brightness invariance. These two feature descriptors are scale-invariant and are very useful for tracking target ships that show a varied appearance in surveillance videos. The Gabor descriptor is insensitive to illumination variance, and we employ a Two-Dimensional (2D) Gabor filter as another distinct ship edge descriptor. More details can be found in Sun et al. (2005). A Canny descriptor is a ladder-shaped edge detector which is sensitive in detecting ship edge variation (Canny, 1986). The proposed ship tracker extracts these five distinct edge descriptors and integrates them into a coupled ship tracking descriptor. In the following section, we use symbol M, instead of five, to indicate the number of views in our proposed multi-view learning model.

2.1.2. *Robust Ship Tracker based on Multi-view learning and Sparse representation (STMS).*

2.1.2.1. *Ship tracking model by multi-view learning and sparse representation.* Hong et al. (2013) and Mei et al. (2015) proposed a generative multi-view learning-based tracker which showed impressive tracking performance. However the tracker proposed by Hong et al. (2013) cannot be used to perform the visual ship tracking task directly. The reason is that the to-be-tracked features in the Hong-proposed tracker are of a colour and intensity which may be quite similar among different ships. Thus, the Hong-proposed tracker is prone to fail in the visual ship tracking task. To address this problem, we propose a robust multi-view learning and sparse representation-based ship tracker.

For a view k in the STMS model, we denote C^k as a complete ship dictionary, and X^k is the corresponding feature matrix. The representation matrix for C^k is demonstrated as W^k. Our tracker solves the ship tracking problem by finding the minimum value of Equation (1). The product of the complete dictionary and feature matrix of M views ($\sum_{k=1}^{M} C^kW^k$

in Equation (1)) represents a potential ship target. $f_L(C^kW^k - X^k)$ (k = 1, 2, ... M) is a cost function of evaluating the vessel reconstruction error between the ship candidate and ship template regarding the k*th* view. A smaller value of the cost function $f_L(C^kW^k - X^k)$ means better tracking performance. Hence, the optimal tracking performance of our tracker is obtained when we find the minimum value of Equation (1).

We introduce a sparse representation method to find a robust solution for Equation (1) more efficiently. We decompose the representation matrix W^k (k = 1, 2...M) as coefficient matrix P^k and Q^k (see Equation (2)). P^k is the representation matrix for the k*th* view. Global representation matrix P (similar to Q) is acquired by padding P^ks (k = 1, 2...M) horizontally. The STMS model of Equation (1) is then rewritten as Equation (3). As the dictionary C^k is an over-complete dictionary in the k*th* view of Equation (3), more elements in P^k and Q^k will better explore the intrinsic relationship between the M-views. Hence, the discrepancy between the ship candidate's feature $C^k(P^k + Q^k)$ and the extracted feature X^k will become smaller. A higher order of P and Q will lead to better tracking accuracy in terms of the cost function in Equation (3).

However, a higher order for the coefficient matrices P and Q results in higher complexity for our tracker. So, the method of group Least Absolute Shrinkage and Selection Operator (LASSO) penalty, denoted as $\ell_{1,2}$, is introduced to reduce the tracker's complexity. Parameter $\|P\|_{1,2}$ represents the independence of each view and is constrained via row sparse representation through the $\ell_{1,2}$ method. The constraint of the row sparse representation for the matrix P helps our tracker explore latent relationships between different views, and a stronger relationship will lead to a lower order of P. Meanwhile, the lower order of P indicates that P is a lower-order sparsity matrix. Parameters λ_1 in Equation (3) control the sparsity of the coefficient matrix of P. The term $\lambda_1\|P\|_{1,2}$ reduces the order of P to a minimum level in the procedure of finding the optimal solution in Equation (3).

$\|Q^T\|_{1,2}$ in Equation (3) represents the abnormal tracking results which are subjected to a group LASSO penalty, and λ_2 is a sparse coefficient of Q. A smaller order for Q means that the tracking results have fewer outliers. Thus, fewer elements in the over-complete dictionary will be employed in the proposed ship tracker. In summary, lower orders for P and Q represent smaller values for the term of $\lambda_1\|P\|_{1,2} + \lambda_2\|Q^T\|_{1,2}$, which leads to a better solution for Equation (3). Based on the above discussions about the constraints on the sparse matrices P and Q, we can conclude that our proposed ship tracker can find the optimal tracking result by solving Equation (3). Previous studies have shown that high performance is obtained by substituting the cost function in Equation (3) with the Frobenius norm (Mei et al., 2015). The Frobenius norm is denoted as $\|\bullet\|_F^2$. Therefore, our tracker can be reformulated to Equation (4).

$$\underset{c,w,x}{\mathrm{Min}} \sum_{k=1}^{M} f_L\left(C^kW^k - X^k\right) \tag{1}$$

$$W^k = P^k + Q^k \quad k = 1, 2 \ldots M \tag{2}$$

$$\underset{c,w,x}{\mathrm{Min}} \sum_{k=1}^{M} f_L\left(C^kW^k - X^k\right) + \lambda_1\|P\|_{1,2} + \lambda_2\|Q^T\|_{1,2} \tag{3}$$

$$\underset{c,w,x}{\mathrm{Min}} \sum_{k=1}^{M} \frac{1}{2}\|C^kW^k - X^k\|_F^2 + \lambda_1\|P\|_{1,2} + \lambda_2\|Q^T\|_{1,2} \tag{4}$$

2.1.2.2. *Optimal solution for STMS model.* The tracking performance of the STMS model largely depends on the solution of Equation (4). For the sake of simplicity, we rewrite Equation (4) into two parts as shown in Equations (5) and (6), respectively. So, solutions for Equations (5) and (6) represent the potential tracking target. The Accelerated Proximal Gradient (APG) method is efficient in solving Frobenius-norm based problems (Zhang et al., 2012). In fact, the APG method acquires $\mathrm{O}\left(\frac{1}{\mathrm{m}^2}\right)$ residual of the optimal solution at the m*th* iteration. Hence, we employ an APG algorithm to solve Equations (5) and (6). The APG algorithm finds an optimal solution for Equation (4) through two steps: composite gradient mapping and aggregation.

$$\mathrm{s}\left(\mathrm{P},\mathrm{Q}\right)=\sum_{\mathrm{k}=1}^{\mathrm{M}}\frac{1}{2}\|\mathrm{C}^{\mathrm{k}}\mathrm{W}^{\mathrm{k}}-\mathrm{X}^{\mathrm{k}}\|_{\mathrm{F}}^{2} \tag{5}$$

$$\mathrm{r}\left(\mathrm{P},\mathrm{Q}\right)=\lambda_1\|\mathrm{P}\|_{1,2}+\lambda_2\|\mathrm{Q}^{\mathrm{T}}\|_{1,2} \tag{6}$$

Step 1: *Composite Gradient Mapping*. Inspired by Gong et al. (2012) and Hong et al. (2013), we apply a composite gradient mapping algorithm to address Equation (4). Thus, based on Equations (5) and (6), Equation (4) can be reformulated as Equation (7). In Equation (7), s(U, V) is the value of s(P, Q) at point (U, V) and s(U, V) is estimated by the first order of the corresponding Taylor expansion. Parameter r(P, Q) in Ω(P, Q; U, V) is a regularised factor. We employ the squared Euclidean distance between (P, Q) and (R, S) to represent the regularisation term.

$$\begin{aligned}\Omega(\mathrm{P},\mathrm{Q};\mathrm{U},\mathrm{V})&=\mathrm{s}(\mathrm{U},\mathrm{V})+\langle\nabla_{\mathrm{U}}\mathrm{s}(\mathrm{U},\mathrm{V}),\mathrm{P}-\mathrm{U}\rangle+\langle\nabla_{\mathrm{V}}\mathrm{s}(\mathrm{U},\mathrm{V}),\mathrm{Q}-\mathrm{V}\rangle\\&\quad+\frac{\sigma}{2}\|\mathrm{P}-\mathrm{U}\|_{\mathrm{F}}^{2}+\frac{\sigma}{2}\|\mathrm{Q}-\mathrm{V}\|_{\mathrm{F}}^{2}+\mathrm{r}(\mathrm{P},\mathrm{Q})\end{aligned} \tag{7}$$

where $\nabla_{\mathrm{U}}\mathrm{s}(\mathrm{U},\mathrm{V})$ and $\nabla_{\mathrm{V}}\mathrm{s}(\mathrm{U},\mathrm{V})$ are partial derivatives of s(U, V) for U and V respectively. The term $\langle\nabla_{\mathrm{U}}\mathrm{s}(\mathrm{U},\mathrm{V}),\mathrm{P}-\mathrm{U}\rangle$ is the inner product of $\nabla_{\mathrm{U}}\mathrm{s}(\mathrm{U},\mathrm{V})$ and P − U. Also, parameter $\langle\nabla_{\mathrm{U}}\mathrm{s}(\mathrm{U},\mathrm{V}),\mathrm{Q}-\mathrm{V}\rangle$ is the inner product of $\nabla_{\mathrm{V}}\mathrm{s}(\mathrm{U},\mathrm{V})$ and Q − V. Parameter σ is a penalty parameter.

Step 2: *Aggregation*. For the m*th* APG iteration process, $(\mathrm{U}^{\mathrm{m}+1},\mathrm{V}^{\mathrm{m}+1})$ is obtained through a linear combination of $(\mathrm{P}^{\mathrm{m}},\mathrm{Q}^{\mathrm{m}})$ and $(\mathrm{P}^{\mathrm{m}-1},\mathrm{Q}^{\mathrm{m}-1})$. Thus, we can update $(\mathrm{U}^{\mathrm{m}+1},\mathrm{V}^{\mathrm{m}+1})$ with Equations (8) and (9) which is termed as the aggregation step. The parameter β_{m} is obtained by Equation (10).

$$\mathrm{U}^{\mathrm{m}+1}=\mathrm{P}^{\mathrm{m}}+\beta_{\mathrm{m}}\left(\frac{1}{\beta_{\mathrm{m}-1}}-1\right)\left(\mathrm{P}^{\mathrm{m}}-\mathrm{P}^{\mathrm{m}-1}\right) \tag{8}$$

$$\mathrm{V}^{\mathrm{m}+1}=\mathrm{Q}^{\mathrm{m}}+\beta_{\mathrm{m}}\left(\frac{1}{\beta_{\mathrm{m}-1}}-1\right)\left(\mathrm{Q}^{\mathrm{m}}-\mathrm{Q}^{\mathrm{m}-1}\right) \tag{9}$$

$$\beta_{\mathrm{m}}=\begin{cases}\dfrac{2}{\mathrm{m}+3} & m>0\\ 1 & \mathrm{m}=0\end{cases} \tag{10}$$

$\mathrm{P}^0,\mathrm{Q}^0,\mathrm{U}^1,\mathrm{V}^1$ are set to zero for initialisation.

Given an aggregation point $(\mathrm{U}^{\mathrm{m}},\mathrm{V}^{\mathrm{m}})$, the solution for the m*th* APG iteration process is obtained by solving Equation (11). The minimisation Equation (11) is reformulated as finding optimal solutions for $\mathrm{P^m}$ and $\mathrm{Q^m}$ in Equations (12) and (13). According to Hong et al. (2013) there is an efficient close-form solution for obtaining an optimal $\mathrm{P^m}$ and $\mathrm{Q^m}$. Equations (14) and (15) present final solutions for $\mathrm{P^m}$ and $\mathrm{Q^m}$, respectively. In Equation (14), $\mathrm{P^m_{i,.}}$ is the i*th* row of $\mathrm{P^m}$, and $\mathrm{Q^m_{.,j}}$ in Equation (15) is the j*th* column for $\mathrm{Q^m}$. $\mathrm{U^m_{i,.}}$ is the i*th* row of $\mathrm{U^m}$ and $\mathrm{V^m_{.,j}}$ is the j*th* column of $\mathrm{V^m}$. We obtain P and Q by computing Equations (14) and (15) iteratively. Our proposed ship tracker determines the final tracked ship based on the final solutions of Equations (14) and (15). The symbol $\|\bullet\|$ in Equation (14) and (15) represents the Euclidean distance.

$$(\mathrm{P^m}, \mathrm{Q^m}) = \arg_{\mathrm{P,Q}} \min \Omega(\mathrm{P}, \mathrm{Q}; \mathrm{U^m}, \mathrm{V^m}) \tag{11}$$

$$\mathrm{P^m} = \arg \underset{P}{\mathrm{Min}} \frac{1}{2} \|\mathrm{P} - (\mathrm{U^m} - \frac{1}{\sigma} \nabla_{\mathrm{U}} \mathrm{s}(\mathrm{U^m}, \mathrm{V^m}))\|_{\mathrm{F}}^{2} + \frac{\lambda_1}{\sigma} \|\mathrm{P}\|_{1,2} \tag{12}$$

$$\mathrm{Q^m} = \arg \underset{Q}{\mathrm{Min}} \frac{1}{2} \left\| \mathrm{Q} - \left(\mathrm{V^m} - \frac{1}{\sigma} \nabla_{V} \mathrm{s}(\mathrm{U^m}, \mathrm{V^m}) \right) \right\|_{F}^{2} + \frac{\lambda_2}{\sigma} \|\mathrm{Q^T}\|_{1,2} \tag{13}$$

$$\mathrm{P^m_{i,.}} = \max \left(0, 1 - \frac{\lambda_1}{\|\sigma \mathrm{U^m_{i,.}} - \nabla_{\mathrm{U}} \mathrm{s}(\mathrm{U^m_{i,.}}, \mathrm{V^m})\|} \right) \mathrm{U^m_{i,.}} \tag{14}$$

$$\mathrm{Q^m_{.,j}} = \max \left(0, 1 - \frac{\lambda_2}{\|\sigma \mathrm{V^m_{.,i}} - \nabla_{\mathrm{V}} \mathrm{s}(\mathrm{U^m}, \mathrm{V^m_{.,i}})\|} \right) \mathrm{V^m_{.,j}} \tag{15}$$

3. EXPERIMENTS AND RESULTS.

3.1. *Data.* Currently, there are few ship video datasets available for evaluating visual ship trackers' performance. Hence, it is necessary to establish a benchmark of ship videos. To this end, we have taken maritime surveillance videos of four common scenarios at Shanghai Port. The collected videos for the four scenarios are denoted as Case-1, Case-2, Case-3 and Case-4. It is known that ripples and wakes impose strong interference on ship tracking models, so Case-1 aims to test the ship tracker's robustness within the constraints.

Another challenge is ship-overlapping in the image sequences, and we collected two videos to evaluate the ship tracker's performance under such a situation. Specifically, Case-2 is a video in which a to-be-tracked ship shows a different appearance (that is, colour and shapes, etc.) from the neighbouring ships. In the other video (that is, Case-3), the to-be-tracked ship has a similar appearance to its neighbours. Case-4 is to assess a tracker's performance in a tracking situation where the to-be-tracked ship is in the distance. To visualise a ship tracker's performance, we have manually labelled the Target Region (TR), in the form of a rectangle, in each frame of the four datasets. The TR rectangle is stored with the information of x-coordinate and y-coordinate of its upper-left vertex, width and height. Ship videos and TR sequences of the above four cases are available from the correspondence e-mail address. As these four cases are common situations in ship tracking applications, they can assist in adequately assessing and evaluating a ship tracker's performance. Samples of the collected videos for each case are shown in Figure 2.

3.2. *Evaluation criteria.* To evaluate our ship tracker's performance, we compared the tracking results with the manually labelled TR rectangles in the collected videos. Tracking results are stored in the same data format as TR rectangles. For simplicity, both the TR rectangle and the tracked ship rectangle are denoted by their centre point (that is, the Intersection Point (IP) of a rectangle's diagonals). Based on that, we compare the

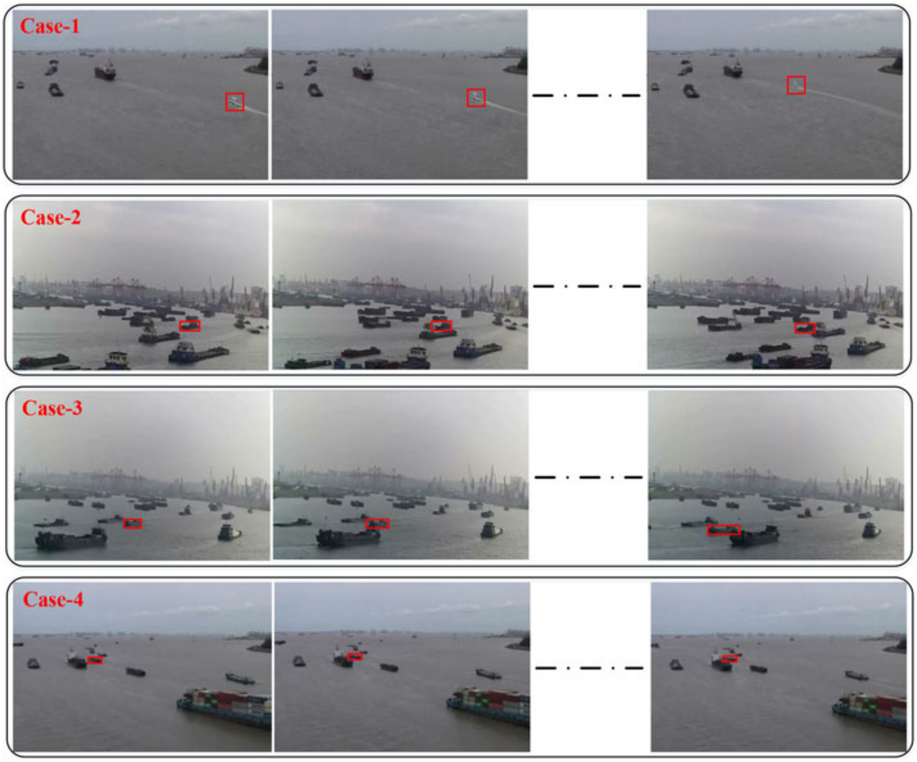


Figure 2. Samples of collected datasets for each case (the TR is encircled with red rectangle in each image).

distance of centre points between the tracked and TR rectangles to quantify a ship tracker's performance.

Two statistical indicators are used to demonstrate a ship tracker's performance, which are Mean Square Error (MSE) and Mean Absolute Difference (MAD). For a given video with n-frames, we denote $\mathrm{G}_{IP}(x,y)$ as the IP coordinate of a rectangle in the TR sequences. $\mathrm{T}_{IP}(x,y)$ is the centre point of the tracked rectangle. Here, the square root is denoted by symbol $\|\bullet\|^{\frac{1}{2}}$ and a quadratic operator is denoted by $\|\bullet\|^2$. So, the offset between the tracked ship-rectangle and the TR rectangle is measured by the distance between $\mathrm{G}_{IP}(x,y)$ and $\mathrm{T}_{IP}(x,y)$ (see Equation (16)). Parameter $\mathrm{D}_t(x)$ represents the squared distance, in the x-axis, between $\mathrm{G}_{IP}(x,y)$ and $\mathrm{T}_{IP}(x,y)$ of the t*th* frame. Similarly, D_t (y) represents the squared distance in the y-axis. Parameters D_t (x) and D_t (y) are obtained by Equations (17) and (18), respectively.

After obtaining the distance in Equation (16), a ship tracker's performance is measured by the statistical indicators MSE and MAD (see Equations (19) and (20)). Parameter $\bar{D}$ in Equation (19) is the average Euclidean distance. Equation (21) presents the calculation process for $\bar{D}$. Smaller values of MSE and MAD show better tracking performance for the ship tracker.

$$D_t(\mathrm{G}_{IP}(x,y), \mathrm{T}_{IP}(x,y)) = \|\mathrm{D}_t(\mathrm{x}) + \mathrm{D}_t(\mathrm{y})\|^{\frac{1}{2}} \tag{16}$$

$$\mathrm{D}_t(\mathrm{x}) = \|G^t_{IP}(x) - T^t_{IP}(x)\|^2 \tag{17}$$

Table 1. Statistical performance of different ship trackers for Case-1.

	MSE (pixels)	MAD (pixels)
Kalman	176·6	142·7
Meanshift	26·9	17·2
STMS	**3·6**	**2·8**

$$\mathrm{D}_t(\mathrm{y}) = \|G^t_{IP}(y) - T^t_{IP}(y)\|^2 \tag{18}$$

$$MSE = \left\| \frac{\sum_{t=1}^{n} \|D_t(\mathrm{G}_{IP}(x,y), \mathrm{T}_{IP}(x,y)) - \bar{D}\|^2}{n} \right\|^{\frac{1}{2}} \tag{19}$$

$$MAD = \frac{\sum_{t=1}^{n} |D_t(\mathrm{G}_{IP}(x,y), \mathrm{T}_{IP}(x,y)) - \bar{D}|}{n} \tag{20}$$

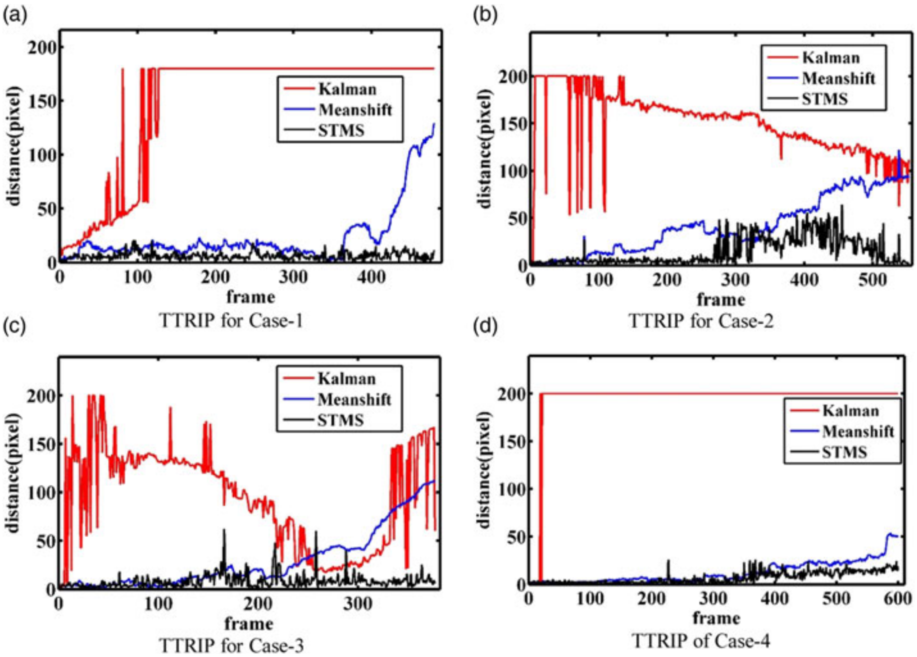
$$\bar{D} = \frac{\sum_{t=1}^{n} D_t(\mathrm{G}_{IP}(x,y), \mathrm{T}_{IP}(x,y))}{n} \tag{21}$$

3.3. *Experiment setup and results.* To evaluate our tracker's performance, two classical tracking algorithms, a basic Kalman tracker (Welch and Bishop, 1995) and a Meanshift tracker (Comaniciu et al., 2000), were implemented on the collected datasets. All trackers were executed on the Windows 10 Operation System PC with 8GB RAM and a 3·4 GHz CPU. Meanwhile, these models were built through Matlab (R2011 version) and a c++ library. The TR locations of the first frame in the four datasets were used to initialise ROI for the three trackers.

3.3.1. *Tracking results for Case-1.* All three trackers have been tested on the Case-1 dataset to present their robustness against ripple interference. It is clearly shown in Table 1 that our proposed ship tracker was better than other trackers. Specifically, MSE for the Kalman tracker was six times larger than that of the Meanshift tracker. Meanwhile, the MSE obtained by the Meanshift tracker was almost eight times larger than the STMS model. In sum, the STMS model could track a ship successfully in terms of MSE, while the Kalman tracker was the worst tracker under Case-1. The minimum statistical tracking errors of MSE and MAD were 3·6 and 2·8 pixels, respectively. Both of these were obtained by the STMS model. Based on the tracking errors, denoted by MSE and MAD, it is reasonable to conclude that our proposed ship tracker is more robust to ripple interference when compared to the other two trackers.

Figure 3(a) shows the distance between the Tracked IP and TR's IP (TTRIP). In order to better visualise the distance variation for Case-1, the maximum value of TTRIP was set to 200 pixels. The distance distribution showed that the Kalman tracker was vulnerable to ripple interference. It is observed that the TTRIP, obtained by the Kalman tracker, increased significantly with the increase of tracking sequences, arriving at the maximum value at the 80th frame for the first time. Although the TTRIP fluctuated significantly in the subsequent 30 frames, it stayed at a maximum value from the 110th frame.

In contrast, both the Meanshift and STMS trackers obtained a smaller TTRIP than that of the Kalman tracker as shown in Figure 3(a). In the first 350 frames, the Meanshift and STMS models had similar tracking result as the TTRIPs were quite close to each other. However, the TTRIP of the Meanshift tracker increased dramatically in the last 130



TTRIP for Case-1

TTRIP for Case-2

TTRIP for Case-3

TTRIP of Case-4

Figure 3. Distance distribution of TTRIP for the three trackers in each case.

frames while STMS maintained a small value. After carefully checking the initial tracking video, we found that the target ship became smaller in the later frames. So, some ripples were tracked by the Meanshift tracker. As ripples randomly changed in each frame, such variation resulted in a degraded performance of the Meanshift tracker. Figure 3(a) shows that the STMS tracker obtained smaller values than the other two trackers for TTRIP in Case-1.

Some typical tracking results of the three trackers for Case-1 are shown in Figure 4. At the beginning of tracking, all three trackers showed good tracking results (see frame # 40 of Figure 4). However, the Kalman and Meanshift trackers failed to track the ship in the latter two frames, as shown in frames # 372, and 450 in Figure 4. The main reason is that the two traditional ship trackers track shallow and single ship features (such as pixel areas moving at limited speed and minimal intensity variations) which are sensitive to interference (such as neighbouring ships' movement, wakes, and own-ship's appearance and size variation in frames). The STMS tracker extracted a set of affine invariant ship features (LBP, HOG, Gabor, LoG and Canny descriptors), and assembled them into a robust distinctive ship feature. The interference feature (that is, wakes feature in the four frames of Figure 4) is completely different from the target ship and thus can be easily suppressed. Based on the aforementioned analysis, we can draw the conclusion that the STMS tracker is robust to wake interference.

3.3.2. *Tracking results for Case-2 and Case-3.* Case-2 involved the image sequences where the target ship was dissimilar to the overlapping ship. Statistical indices listed in Table 2 show that the Kalman tracker achieved the worst tracking performance and the STMS model continued to achieve the best tracking performance. The minimum MSE, in

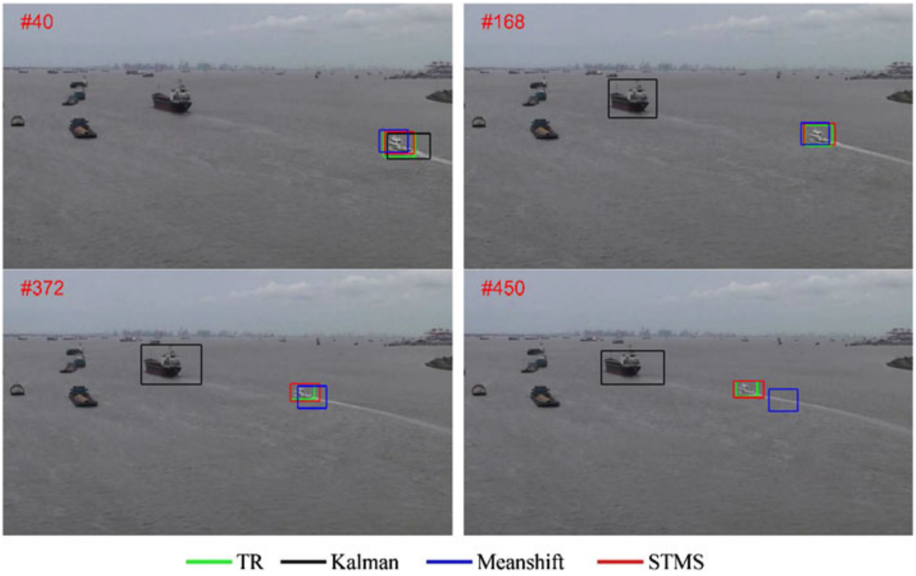


Figure 4. Tracking results for different trackers of Case-1.

Table 2. Statistical performance of different ship trackers for Case-2.

	MSE (pixels)	MAD (pixels)
Kalman	57·9	38·0
Meanshift	29·5	24·8
STMS	**14·2**	**12·0**

Case-2, was 14·2 pixels, which was obtained by the STMS tracker. MSEs for the Kalman and Meanshift trackers were at least one-fold higher than that of the STMS model. The MAD indicator for Case-2 presented a similar variation trend as MSE. The minimum value for MAD was 12 pixels which was obtained by the STMS model. From the perspective of MSE and MAD, our proposed tracker achieved the best performance.

Figure 3(b) shows TTRIP distributions for case-2. We found that tracking accuracy for the Kalman model fluctuated significantly in the first 200 frames. After carefully checking the tracking results, we observed that the Kalman tracker tracked almost one-third of each frame's image patch. However, the TTRIP of the Kalman tracker showed a decreasing trend in the last 350 frames. The reason is that the target ship sailed towards the Kalman's tracked ROI in the last 350 image sequences. The Meanshift tracker lost the target in the 200th frame and then its tracking ROI located the overlapped ship, hence the corresponding TTRIP for the Meanshift tracker experienced an increasing tendency as shown in Figure 3(b).

The Kalman tracker considered the region with large and obvious displacement as the target ship. Thus, three moving ships and neighbouring wake pixels were wrongly tracked as the target ship as shown in frame # 74 of Figure 5. Due to the same reason, the Kalman tracker lost the target ship in frames # 462 and # 540 as shown in Figure 5. We also observed

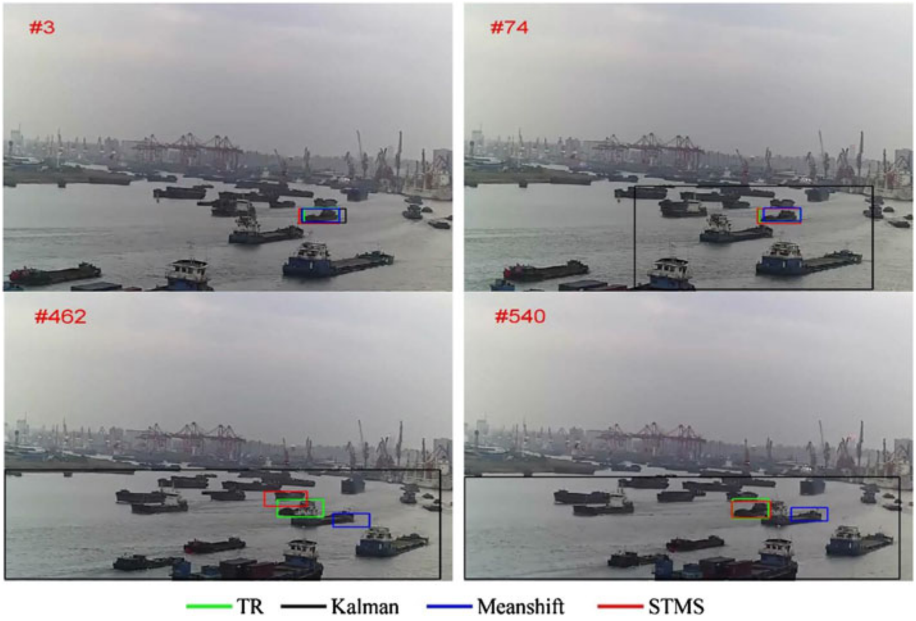


Figure 5. Tracking results for different trackers for Case-2.

Table 3. Statistical performance of different ship trackers for Case-3.

	MSE (pixels)	MAD (pixels)
Kalman	52·9	45·5
Meanshift	30·2	24·1
STMS	**7·3**	**4·8**

that the Meanshift tracker failed to track the target ship when the target ship was sheltered by its neighbour (that is, in frame # 462 and # 540 in the same figure). The reason is that only the tracked target feature in the previous frame was employed to help the Meanshift tracker determine the candidate ship area. The advantage of the STMS tracker is that it persistently updated its tracking template. Thus, although the STMS tracker lost the target when the target was sheltered by its neighbour (see frame # 462 in Figure 5), the STMS tracker could still determine the tracking ship correctly as long as the ship re-appeared in the frame (see frame # 540 in Figure 5).

Case-3 tested the robustness of three trackers where the target ship's appearance was similar to its overlapping neighbour. Overall, tracking accuracy for the three trackers was better than Case-2 in terms of MSE and MAD. As shown in Table 3, the MSE obtained by the STMS tracker was 7·3 pixels which is less than a quarter of that of the Meanshift tracker. In addition, the MSE of the Kalman tracker was 52·9 pixels which was almost twice as high as the Meanshift tracker. The variation trend of MAD was consistent with MSE in Case-3. Distributions of statistical indicators showed that the STMS model obtained better tracking accuracy compared with the Kalman and Meanshift trackers.

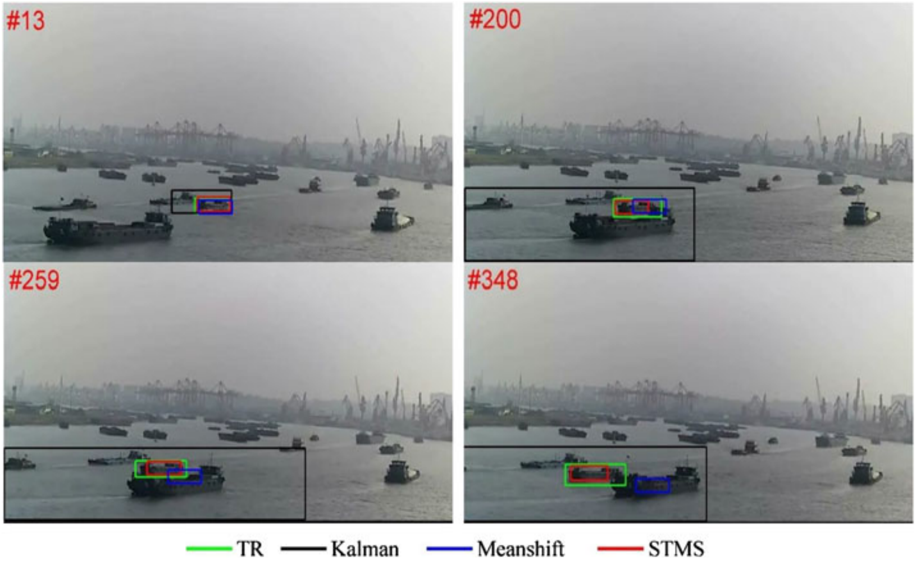


Figure 6. Tracking results for different trackers for Case-3.

TTRIP, shown in Figure 3(c), showed that the magnitude of tracking errors for both Kalman and Meanshift trackers were lower in contrast to Case-2. In addition, there were some peaks observed in TTRIP of the STMS tracker. This phenomenon showed that the STMS model failed to distinguish the target ship and the overlapped ship when they encountered each other in the video. However, in contrast to the Kalman and Meanshift trackers, the STMS re-found the target ship in later frames.

Ship tracking in Case-3 was more difficult than in Case-2 in that the overlapping ship imposed additional interference to the different trackers. As ships in the bottom-left area (see frames #200, # 259, #348 in Figure 6) had less observable movements than other ships in Case-3, the Kalman tracker mistook the bottom-left area as the target ship region. At the beginning of overlap (see frame # 200 in Figure 6), a partial target ship was visible, and the Meanshift tracker could still track the target ship despite a large tracking error. When the target ship was completely sheltered by its neighbour, the Meanshift tracker failed to find the target ship's feature, and could only track its neighbour (see frames # 259 and # 348 in Figure 6). In contrast, the STMS tracker, with the ability of updating the tracking template automatically and extracting highly-coupled ship features, could track the target ship robustly when the target ship was visible ((see frames # 200, # 259, # 348 in the same figure). With the above analysis, we can say that the STMS tracker can track ships successfully under an overlapping situation.

3.3.3. *Tracking results for Case-4.* The last video aims to test different trackers' performance on tracking ships in the distance. As shown in Table 4, the MSE for the Kalman tracker was 70·9 pixels, which is larger than the MSE values of the Meanshift and STMS trackers. This shows that the Kalman tracker was not good at tracking ships in this situation. Compared to the Kalman tracker, the indicators in Table 4 show that the Meanshift tracker has a much closer tracking accuracy to the STMS tracker than the Kalman tracker. The MAD for the Kalman tracker was 33·6 pixels while its counterpart for the STMS tracker

Table 4. Statistical performance of different ship trackers for Case-4.

	MSE (pixels)	MAD (pixels)
Kalman	70·9	33·6
Meanshift	11·5	9·4
STMS	**5·9**	**5·1**

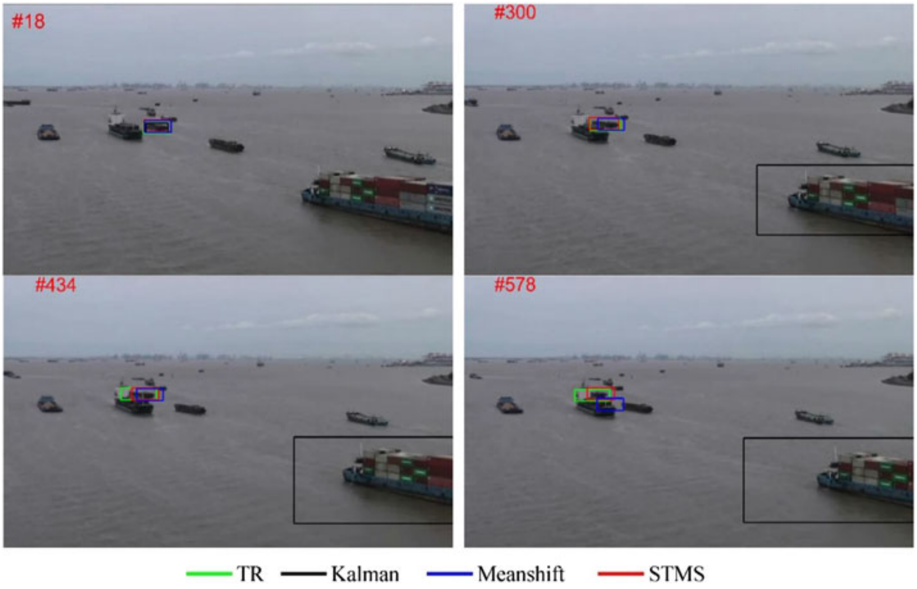


Figure 7. Tracking results for different trackers for Case-4.

was 5·1 pixels. MAD for the Meanshift model was almost twice that of the STMS model. From the perspective of MSE and MAD, the STMS tracker performed better than the Kalman and Meanshift trackers in Case-4. The TTRIP distributions shown in Figure 3(d) proves our analysis.

The main challenge of ship tracking in Case-4 is that the target ship's visual contours are blurred, and thus it is difficult to extract distinctive features for tracking ships. In this case, the Kalman tracker was interfered with by the bottom-right container ship in frames # 300, 434, and 578 in Figure 7. Meanwhile, the Meanshift tracker lost the target in the frame # 578 in Figure 7 as the tracked area (that is, the wrongly tracked ship) had more clear and distinctive colour information. The STMS tracker could extract the target ship's highly-coupled feature, and obtained robust tracking results (see frames # 300, 434, and 578 in Figure 7). Based on the above discussions, our proposed STMS ship tracker can obtain robust and reliable tracking results under different tracking scenarios.

4. CONCLUSION.

Visual ship tracking is a key technology to support the development of smart shipping, but to be successful, it must overcome many challenges and different situations. Here, we propose a robust visual ship tracking framework via a multi-view learning mechanism and a sparse representation method to solve the problem. The proposed ship tracker employs edge information-based features (HOG, LBP, Gabor filter, LoG and Canny descriptors) to find a target ship in tracking videos. We have assessed our trackers' performance in four challenging tracking situations. Experimental results show that the proposed visual ship tracking model can achieve great success. From the perspective of MAD and MSE, our proposed tracker obtained better tracking performance compared to traditional object tracking models such as Kalman and Meanshift trackers. Specifically, the location of tracked ships obtained from the STMS tracker was very close to the manually labelled ROI in terms of TTRIP. The proposed visual ship tracking model offsets some of the disadvantages of AIS, SAR, LRIT and other traditional tracking technologies. Thus, the combination of the proposed visual tracking method and traditional tracking technologies could help a smart ship perceive the navigation environments in a more timely and efficient manner. This has the potential to assist in the improvement of maritime traffic safety in the smart ship era.

In future, we will focus on two aspects. First, the proposed method can obtain satisfactory tracking accuracy for single ship tracking tasks. Testing the proposed model's performance on tracking multiple ships is now required. Second, as some tracking outliers still exist in ship-intersection scenarios, we will employ more traditional ship tracking techniques (such as AIS, SAR, LRIT) to increase the tracking accuracy of the proposed visual tracking model.

ACKNOWLEDGEMENTS

This research is supported by Shanghai Shuguang Plan Project (No:15SG44), National Natural Science Foundation of China (No: 51379121, 51709167, 51579143, 61304230), China Postdoctoral Science Foundation (No: 2015M581585) and Shanghai Committee of Science and Technology, China (No: 18040501700).

REFERENCES

Allen, J. G., Xu, R. Y. and Jin, J. S. (2004). Object tracking using camshift algorithm and multiple quantized feature spaces. *Proceedings of the Pan-Sydney area workshop on Visual information processing, Australian Computer Society*, **36**, 3–7.

Bardow, P., Davison, A. J., and Leutenegger, S. (2016). Simultaneous optical flow and intensity estimation from an event camera. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 884–892.

Burmeister, H. C., Bruhn, W., Rødseth, Ø. J., and Porathe, T. (2014). Autonomous Unmanned Merchant Vessel and its Contribution towards the e-Navigation Implementation: The MUNIN Perspective. *International Journal of e-Navigation and Maritime Economy*, **1**, 1–13.

Canny, J. (1986). A computational approach to edge detection. *IEEE Transactions on pattern analysis and machine intelligence*, **PAMI-8**(6), 679–698.

Chaturvedi, S. K., Yang, C. S., Ouchi, K., and Shanmugam, P. (2012). Ship recognition by integration of SAR and AIS. *The Journal of Navigation*, **65**(02), 323-337.

Chauhan, A. K. and Krishan, P. (2013). Moving object tracking using gaussian mixture model and optical flow. *International Journal of Advanced Research in Computer Science and Software Engineering*, **3**(4), 243–246.

Chen, Y. (2014). Satellite-based AIS and its Comparison with LRIT. *Transnav International Journal on Marine Navigation & Safety of Sea Transportation*, **8**(2), 183–187.

Comaniciu, D., Ramesh, V., and Meer, P. (2000). Real-time tracking of non-rigid objects using mean shift. *IEEE Conference on Computer Vision and Pattern Recognition*, **2**, 142–149.

Dalal, N., and Triggs, B. (2005). Histograms of oriented gradients for human detection. *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, **1**, 886–893.

Delamarre, Q. and Faugeras, O. (1999). 3D articulated models and multi-view tracking with silhouettes. *Proceedings of the Seventh IEEE International Conference on Computer Vision*, **2**, 716–721.

Friedman, J., Hastie, T. and Tibshirani, R. (2008). Sparse inverse covariance estimation with the graphical lasso. *Biostatistics*, **9**(3), 432–441.

Gong, P., Ye, J., and Zhang, C. (2012). Robust multi-task feature learning. *Proceedings of the 18th ACM SIGKDD international conference on Knowledge discovery and data mining*, 895–903.

Gray, G. J., Aouf, N., Richardson, M. A., Butters, B. and Walmsley, R. (2012). An Intelligent tracking algorithm for an imaging infrared anti-ship missile. *SPIE Security+ Defence, International Society for Optics and Photonics*, **8543**, 85430L.

Gray, G. J., Aouf, N., Richardson, M. A., Butters, B., Walmsley, R. and Nicholls, E. (2011). Feature-based tracking algorithms for imaging infrared anti-ship missiles. *SPIE Security+ Defence, International Society for Optics and Photonics*, **8187**, 81870T.

Gunn, S. R. (1999). On the discrete representation of the Laplacian of Gaussian. *Pattern Recognition*, **32**(8), 1463–1472.

Han, S. K., Ra, W. S., Whang, I. H. and Park, J. B. (2014). Linear recursive passive target tracking filter for cooperative sea-skimming anti-ship missiles. *IET Radar, Sonar & Navigation*, **8**(7), 805–814.

Hong, Z., Mei, X., Prokhorov, D. and Tao, D. (2013). Tracking via Robust Multi-task Multi-view Joint Sparse Representation. *IEEE Conference on Computer Vision*, 649–656.

Hong, Z., Chen, Z., Wang, C., Mei, X., Prokhorov, D. and Tao, D. (2015). Multi-store tracker (muster): A cognitive psychology inspired approach to object tracking. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 749–758.

Hu, W. C., Yang, C. Y. and Huang, D. Y. (2011). Robust real-time ship detection and tracking for visual surveillance of cage aquaculture. *Journal of Visual Communication and Image Representation*, **22**(6), 543–556.

Joshi, K. A. and Thakore, D. G. (2012). A survey on moving object detection and tracking in video surveillance system. *International Journal of Soft Computing and Engineering*, **2**(3), 44–48.

Lapinski, A. L. S., Isenor, A. W. and Webb, S. (2016). Simulating Surveillance Options for the Canadian North. *The Journal of Navigation*, **69**(5), 940–954.

Loomans, M. J., de With, P. H. and Wijnhoven, R. G. (2013). Robust automatic ship tracking in harbours using active cameras. *2013 IEEE International Conference on Image Processing*, 4117–4121.

Ma, H., Shen, R. and Wu, Z. (2011). The Ship Automatic Tracking Technique and Implementation Based on VTS Images. *ICTIS 2011: Multimodal Approach to Sustained Transportation System Development: Information, Technology, Implementation, ASCE*, 2616–2623.

Mei, X. and Ling, H. (2011). Robust visual tracking and vehicle classification via sparse representation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **33**(11), 2259–2272.

Mei, X., Hong, Z., Prokhorov, D. and Tao, D. (2015). Robust Multitask Multiview Tracking in Videos. *IEEE transactions on Neural Networks and Learning Systems*, **26**(11), 2874–2890.

Silvas, E., Hereijgers, K., Peng, H., Hofman, T. and Steinbuch, M. (2016). Synthesis of Realistic Driving Cycles With High Accuracy and Computational Speed, including Slope Information. *IEEE Transactions on Vehicular Technology*, **65**(6), 4118–4128.

Statheros, T., Howells, G. and Maier, K. M. (2008). Autonomous ship collision avoidance navigation concepts, technologies and techniques. *The Journal of Navigation*, **61**(01), 129–142.

Suard, F., Rakotomamonjy, A., Bensrhair, A. and Broggi, A. (2006). Pedestrian detection using infrared images and histograms of oriented gradients. *IEEE Intelligent Vehicles Symposium*, 206–212.

Sun, Z., Bebis, G. and Miller, R. (2005). On-Road Vehicle Detection Using Evolutionary Gabor Filter Optimization. *IEEE Transactions on Intelligent Transportation Systems*, **6**(2), 125–137.

Szpak, Z. L. and Tapamo, J. R. (2011). Maritime surveillance: Tracking ships inside a dynamic background using a fast level-set. *Expert Systems with Applications*, **38**(6), 6669–6680.

Taj, M. and Cavallaro, A. (2010). Multi-view multi-object detection and tracking. *In Computer Vision, Springer, Berlin, Heidelberg*, **285**, 263–280.

Tang, J., Wang, Y. and Liu, F. (2013). Characterizing traffic time series based on complex network theory. *Physica A Statistical Mechanics & its Applications*, **392**(18), 4192–4201.

Tang, J., Zhang, G., Wang, Y., Wang, H. and Liu, F. (2015). A hybrid approach to integrate fuzzy C-means based imputation method with genetic algorithm for missing traffic volume data estimation. *Transportation Research Part C Emerging Technologies*, **51**, 29–40.

Tang, J., Liu, F., Zou, Y., Zhang, W. and Wang, Y. (2017). An Improved Fuzzy Neural Network for Traffic Speed Prediction Considering Periodic Characteristic. *IEEE Transactions on Intelligent Transportation Systems*, **18**(9), 2340–2350.

Teng, F. and Liu, Q. (2014). Robust Multiple Ship Tracking in Inland Waterway CCTV System. *Indonesian Journal of Electrical Engineering and Computer Science*, **12**(11), 7772–7777.

Teng, F., Liu, Q. and Zhu, L. (2014). Robust Inland Waterway Ship Tracking via Orthogonal Particle Filter-based Predator. *International Journal of Signal Processing, Image Processing and Pattern Recognition*, **7**(6), 125–136.

Teng, F., Liu, Q., Gao, X. Y. and Zhu, L. (2013). Real-time ship tracking via enhanced MIL tracker. *IETET, Conference Publishing System, Kurukshetra, India*, 399–404.

Tripathi, R. P., Ghosh, S. and Chandle, J. O. (2016). Tracking of object using optimal adaptive Kalman filter. *IEEE International Conference on Engineering and Technology (ICETECH)*, 1128–1131.

Vespe, M., Greidanus, H. and Alvarez, M. A. (2015). The declining impact of piracy on maritime transport in the Indian Ocean: Statistical analysis of 5-year vessel tracking data. *Marine Policy*, **59**, 9–15.

Wang, N. (2010). An intelligent spatial collision risk based on the quaternion ship domain. *The Journal of Navigation*, **63**(04), 733–749.

Welch, G. and Bishop, G. (1995). An introduction to the Kalman filter. Technical Report TR 95-041, Department of Computer Science, University of North Carolina.

Xiao, F., Ligteringen, H., Van Gulijk, C. and Ale, B. (2015). Comparison study on AIS data of ship traffic behavior. *Ocean Engineering*, **95**, 84–93.

Xu, C., Tao, D. and Xu, C. (2013). A survey on multi-view learning. *Neural Computing and Applications*, **23**(7–8), 2031–2038.

Yan, Y., Zhang, S., Tang, J. and Wang, X. (2017). Understanding characteristics in multivariate traffic flow time series from complex network structure. *Physica A Statistical Mechanics & its Applications*, **477**, 149–160.

Yang, J., Liang, W. and Jia, Y. (2012). Face pose estimation with combined 2D and 3D HOG features. *IEEE 21st International Conference on Pattern Recognition (ICPR)*, 2492–2495.

Zhang, S., Tang, J., Wang, H., Wang, Y. and An, S. (2017b). Revealing intra-urban travel patterns and service ranges from taxi trajectories. *Journal of Transport Geography*, **61**, 72–86.

Zhang, T., Ghanem, B., Liu, S. and Ahuja, N. (2012). Robust visual tracking via multi-task sparse learning. *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2042–2049.

Zhang, W., Goerlandt, F., Kujala, P. and Wang, Y. (2016). An advanced method for detecting possible near miss ship collisions from AIS data. *Ocean Engineering*, **124**, 141–156.

Zhang, W., Goerlandt, F., Montewka, J. and Kujala, P. (2015). A method for detecting possible near miss ship collisions from AIS data. *Ocean Engineering*, **107**, 60–69.

Zhang, W., Kopca, C., Tang, J., Ma, D. and Wang, Y. (2017a). A systemic approach for near miss ship collision detection. *The Journal of Navigation*, **70**(5), 1117–1132.

Zhao, Z., Ji, K., Xing, X., Zou, H. and Zhou, S. (2014). Ship Surveillance by Integration of Space-borne SAR and AIS–Further Research. *The Journal of Navigation*, **67**(02), 295–309.