Multiple Object Tracking Applies on Ship: A Survey

RONGFENG WEI, City University of Hong Kong

1 MOTIVATION

Multiple object tracking (MOT) is a common computer vision task that requires detecting objects in consecutive video frames and assigning each object a track id, which is unique in the video sequence[10]. In the earlier works, some of them formulated instance association as a graph-based optimization problem under the "tracking-by-detection" paradigm. In practice, the graph-based approach usually requires an expensive computation.

Recently, online trackers are merging, they focus on enhancing the performance of association with each object from different frames[9]. Among these, some works are based on traditional computer vision approaches like Kalman filter and Hungarian algorithm, which advantaged in higher speed, but limited in lower accuracy and frequent id-switch[4]. With the rapid development of deep learning, some excellent works have pushed online MOT into state-of-the-art territory, making them very competitive.

On the other hand, multiple ship tracking (MST) is a sub-task of MOT, which plays an vital role in marine surveillance and ship situational awareness systems. Compared with MOT, the research of MST is much less popular. Lack of works focuses on MST leaving a gap in studies from MOT 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[11]. This proposal aims to utilize and compare the advanced algorithms from MOT to explore a state-of-the-art approach in MST areas.

2 RELATED WORKS

2.1 SORT

2.1.1 Introduction. Simple Online and Realtime Tracking (SORT) [2] is a online tracking framework proposed by Alex Bewley et al. in 2016. The simple tracking method allows SORT to associate objects online effectively and in real-time, which makes SORT outperform other multi-object trackers.

2.1.2 Method. The flowchart of SORT is shown in Fig. 1.

1) The detector gets the detection object
2) Kalman Filter Predict

   SORT uses a linear constant velocity model to describe the state of each target. SORT uses an 8-dimensional vector $(u, v, s, r, \dot{u}, \dot{v}, \dot{s})$ to represent the state space, Where u and v represent the center coordinate of the target, s represents the area of the bounding box, and r represents the aspect ratio. In SORT, r is a constant. The $\dot{v}$ represents the speed, which is solved by the Kalman filter.
3) IOU Match

   The distance between the tracked object bounding box and the detected object bounding box is calculated using the intersection-over-union (IOU) matrix. In order to minimize the IOU distance between the tracking object bounding box and the detection object bounding box, the Hungarian algorithm is used for optimal matching to complete data association.
4) Kalman Filter Update

   When the detection result of the next frame is associated with the tracked object, the result is used as the observation value to update the state of the object in the next frame.

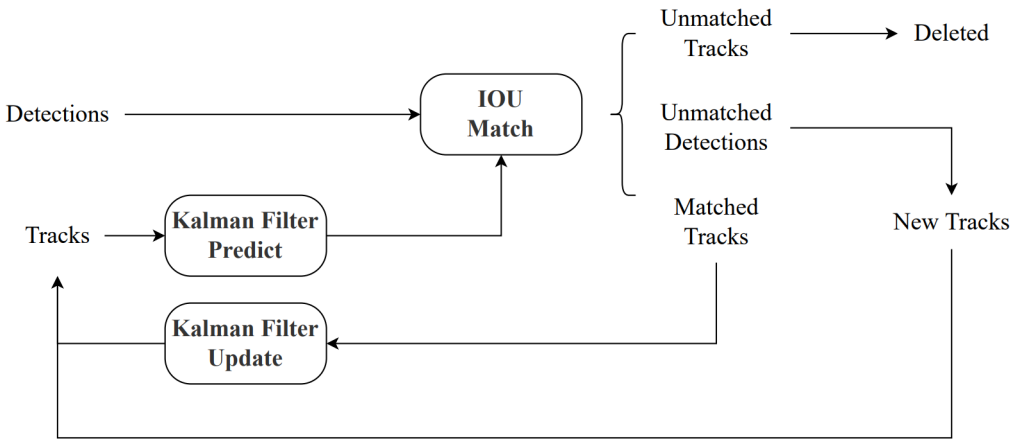

Fig. 1. *The flowchart of the SORT algorithm*

2.1.3 Conclusion. The principle of the SORT algorithm is simple, easy to implement, and maintains accuracy while achieving real-time performance. But there are also shortcomings: SORT does not consider obstructions, whether it is a long or a short time, so in the case of obstructions, the accuracy of SORT is very low.

2.2 DeepSORT

2.2.1 Introduction. In SORT, matching only by IOU is very fast, but the ID switch remains large. This led to the proposal of the DeepSORT [9] algorithms, the biggest feature of DeepSORT is adding appearance information, borrowing the ReID domain model to extract features, and reducing the number of ID switches.

2.2.2 Method. The pipeline of the DeepSORT algorithm is basically the same as SORT, except that there are more Matching Cascade and confirmed.

1) State Estimate

   DeepSORT continues the algorithm of SORT by using an 8-dimensional state space$(u, v, r, h, \dot{x}, \dot{y}, \dot{r}, \dot{h})$. Using the standard Kalman filter with constant velocity motion and linear observation model.

2) Matching Cascade

   As is shown in the Fig. 2, it shows clearly how to do cascade matching. In the upper part, appearance model(ReID) and motion model(Mahalanobis distance) are used to calculate the similarity and obtain the cost matrix. The second part is the data association step of cascade matching. The matching process is a cycle, the trace that has not been lost will be matched first, and the ones that are lost long ago will be matched later.

3) Performance Characteristics

   This part of the apparent feature borrows the network model in the field of pedestrian re-identification. This part of the network needs to be learned offline in advance, and its function is to extract features with a degree of discrimination.

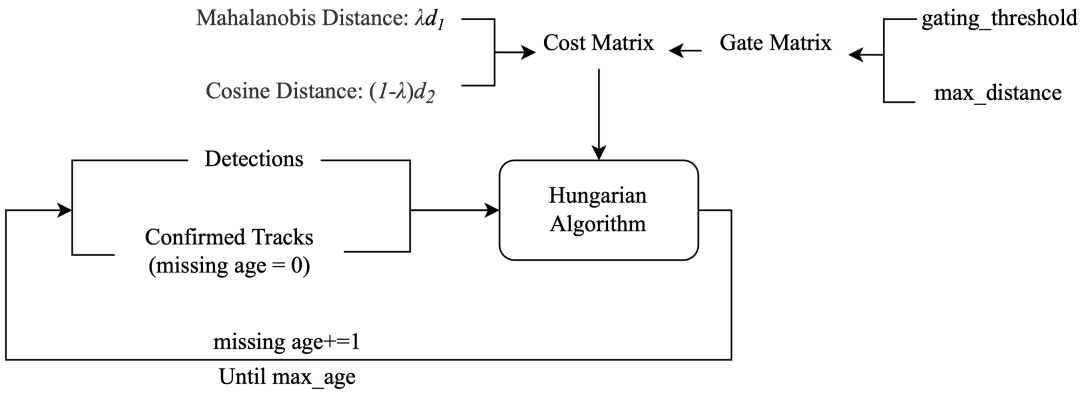

Fig. 2. *The flowchart of Cascade Matching*

2.2.3 Conclusion. DeepSORT utilize the re-id model and the cascade matching to optimize the performance of association in tracking, which has achieved a great success. However,the metrics show not-so-good results. DeepSORT has many drawbacks like ID switches, bad occlusion handling, motion blur, and many more.

2.3 MOTDT

2.3.1 Introduction. MOTDT [6] improves the accuracy of Tracking while ensuring Online-tracking. The core idea is to generate object candidates (bbox) from object detection and object tracking at the same time, and design a new scoring mechanism to select the final candidates. For example, the high confidence result in detection can prevent tracking drifts, and tracking can reduce accidental inaccuracies caused by detection.

2.3.2 Method.

1) Candidate selection

   A unified scoring function is proposed: obtained by object classifier and tracklet confidence. Then use NMS to process all candidate scores to remove redundant candidates.

2) Data association

   Use Appearance representations (Person ReID) and spatial information to associate existing tracks with candidates.

3) real-time object classification

   Using R-FCN to classify the target.Each image frame first passes through the Encoder-Decoder network structure to generate classified Score maps. Define each candidate area as an ROI (represented by x, define width and height as w, h).

4) Tracklet confidence scoring function

   Tracklet confidence measures the accuracy of the filter with timing information. The trajectory is generated by correlating the candidates of successive frames. A track can be split into multiple track segments (Tracklets). The Kalman filter only utilizes the last trajectory segment of a trajectory.

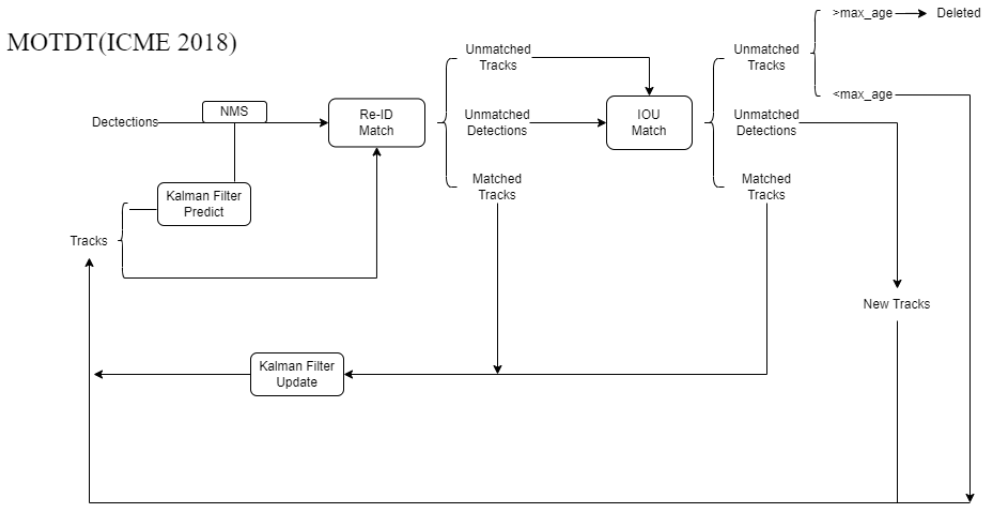

Fig. 3. *The flowchart of the MOTDT algorithm*

3 METHOD

Multiple Object Tracking(MOT) task is to detect all targets on each frame and correlate them across frames in time to form a trajectory. Some early work has seen data correlation as a graph optimization problem under the TBD paradigm, where a node represents a detection box and edge encodes the likelihood of two nodes linking together. In fact, these methods often use a combination of visual cues and motion cues to represent a node, which usually requires a relatively large amount of computation. Moreover, they usually build a large offline graph, and solving on this graph is not easy, which limits the possibility of such methods in real-time tracking. Exploring the importance of motion modeling in a series of online multi-object tracking methods based on SORT. In SORT, a better motion model is the key to improving tracking accuracy, the original SORT uses Kalman filtering based on simple geometric features for motion modeling, while some recent SOTA methods learn a deep network to predict displacement based on visual and geometric features, which greatly improves the accuracy of SORT.

3.1 SiamMOT

Using a region-based twin multi-target tracking network for the exploration of motion modeling, calling it SiamMOT[8]. The authors combine a region-based detection network (Faster R-CNN) and two kinematic models derived from twin single-target tracking (implicit motion model (IMM) and explicit motion model (EMM), respectively). Unlike CenterTrack's implicit target motion prediction based on point-based features, SiamMOT uses region-based features and has developed an explicit mask matching strategy to estimate the motion of the template, which is more robust in challenging scenarios, such as high-speed motion scenarios.

SiamMOT builds upon Faster-RCNN object detector, which consists of a Region Proposal Network(RPN) and a region-based detection network. On top of the standard Faster-RCNN, SiamMOT adds a region-based Siamese tracker to model instance-level motion.SiamMOT takes as input two frames $\mathbf{I}^t, \mathbf{I}^{t+\delta}$ together with a set of detected instances $\mathbf{R}^t = \{R_1^t, \dots R_i^t, \dots\}$ at time t. In SiamMOT, the detection net-work outputs a set of detected instances $\mathbf{R}^{t+\delta}$, while the tracker propagates $\mathbf{R}^t$ to time $t+\delta$ to generate $\tilde{\mathbf{R}}^{t+\delta}$. Finally, the target on the t frame is matched with the prediction box on the $t+\delta$ frame and the detection box on the $t+\delta$ frame, so as to correlate to form a trajectory. Much of the previous work has generally used two frames of features into the network to achieve predictions from R_i^t to $\tilde{\mathbf{R}}^{t+\delta}$, so they are implicitly modeled instance motions. However, many studies of single-object tracking have shown that fine-grained space-level supervision is important for explicitly learning a robust target-matching function in challenging scenarios. Therefore, the authors propose two different twin trackers, one with an implicit motion model and one with an explicit motion model

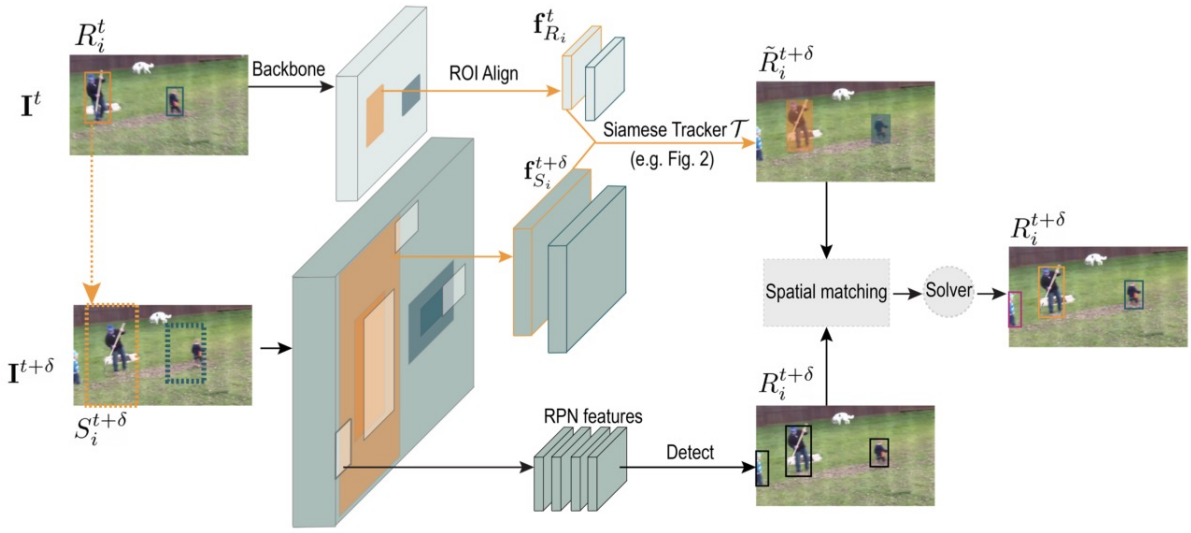

Fig. 4. *(Best viewed in color) SiamMOT is a region-based multi-object tracking network that detects and associates object instances simultaneously. The Siamese tracker models the motion of instances across frames and it is used to temporally link detection in online multi-object tracking. Backbone feature map for frame* $\mathbf{I}^t$ *is visualized with 1/2 of its actual size.*

3.2 Implicit motion model

Implicit motion model (IMM), which uses MLP to estimate the motion between two frames of a target, see as the figure below. Specifically, it first connects features $\mathbf{f}_{R_i}^t$ and $\mathbf{f}_{S_i}^{t+\delta}$ together by channels and then feeds them into MLP to predict the visible confidence level v_i and relative position and scale shift, as shown in the following formula, where $\left(x_i^t, y_i^t, w_i^t, h_i^t\right)$ is the four values of the target box, and through R_i^t and m_i we can easily solve $\tilde{\mathbf{R}}^{t+\delta}$.

$$m_i = \left[\frac{x_i^{t+\delta} - x_i^t}{w_i^t}, \frac{y_i^{t+\delta} - y_i^t}{h_i^t}, \log\frac{w_i^{t+\delta}}{w_i^t} \log\frac{h_i^{t+\delta}}{h_i^t}\right]$$

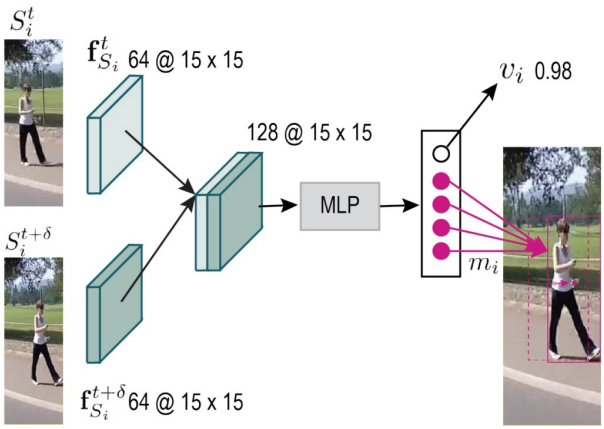

Fig. 5. *Network architecture of Implicit Motion Model (IMM)*

Loss. Given $\left(R_i^t, S_i^{t+\delta}, R_i^{t+\delta}\right)$, IMM training can be carried out using the following loss formula, where v_i^* and m_i^* are calculated according to the G_t label of $R_i^{t+\delta}$, $\mathbb{1}$ is the indication function, ℓ_{focal} is the categorical loss, and ℓ_{reg} is the commonly used smooth ℓ_1

loss for regression.

$$\mathbf{L} = \ell_{\text{focal}}\ \left(v_i, v_i^*\right) + \mathbb{1}\left[v_i^*\right] \ell_{\text{reg}}\ \left(m_i, m_i^*\right)$$

3.3 Explicit motion model

Inspired by the literature on single-object tracking, we propose an explicit motion model(EMM, see in the Fig.6) in SiamMOT. Specifically, it uses a channel-wise cross-correlation operator () to generate a pixel-level response map $\mathbf{r}_i$, which has shown to be effective in modelling dense optical flow estimation and in SOT for instance-level motion estimation. In SiamMOT, this operation correlates each location of the search feature map $\mathbf{f}_{S_i}^{t+\delta}$ with the target feature map $\mathbf{f}_{R_i}^{t}$ to produce $\mathbf{r}_i = \mathbf{f}_{S_i}^{t+\delta} * \mathbf{f}_{R_i}^{t}$, so each map $\mathbf{r}_i[k, :, :]$ captures a different aspect of similarity. Inspired by FCOS, EMM uses a fully convolutional network ψ to detect the matched instances in $\mathbf{r}_i$. Specifically, ψ predicts a dense visibility confidence map $\mathbf{v}_i$ indicating the likelihood of each pixel to contain the target object, and a dense location map $\mathbf{p}_i$ that encodes the offset from that location to the top-left and bottom-right bounding box corners. Thus, we can derive the instance region at (x, y) by the following transformation $\mathcal{R}(\mathbf{p}(x, y)) = [x - l, y - t, x + r, y + b]$ in which $\mathbf{p}(x, y) = [l, t, r, b]$ (the top-left and bottom-right corner offsets). Finally, we decode the maps as follows:

$$\tilde{R}_i^{t+\delta} = \mathcal{R}\left(\mathbf{p}_i\left(x^*, y^*\right)\right); \quad v_i^{t+\delta} = \mathbf{v}_i\left(x^*, y^*\right)$$
$$\text{s.t. } \left(x^*, y^*\right) = \underset{x,y}{\text{argmax}}\left(\mathbf{v}_i \odot \boldsymbol{\eta}_i\right)$$

where $\odot$ is the element-wise multiplication, $\boldsymbol{\eta}_i$ is a penalty map that specifies a non-negative penalty score for the corresponding candidate region as follows:

$$\boldsymbol{\eta}_i(x, y) = \lambda C + (1 - \lambda)\mathcal{S}\left(\mathcal{R}(\mathbf{p}(x, y)), R_i^t\right)$$

where λ is a weighting scalar $\lambda (0 \leq \lambda \leq 1)$, C is the cosinewindow function w.r.t the geometric center of the previous target region $\mathcal{R}_i^t$ and $\mathcal{S}$ is a Guassian function w.r.t the relative scale (height / width) changes between the candidate region $(\mathbf{p}(x, y)))$ and R_i^t. The penalty map $\boldsymbol{\eta}_i$ is introduced to discourage dramatic movements during the course of tracking.

Loss. Given a triplet $\left(R_i^t, S_i^{t+\delta}, R_i^{t+\delta}\right)$, we formulate the training loss of EMM as follows:

$$\mathbf{L} = \sum_{x,y} \ell_{\text{focal}}\ \left(\mathbf{v}_i(x, y), \mathbf{v}_i^*(x, y)\right) + \sum_{x,y} \mathbb{1}\left[\mathbf{v}_i^*(x, y) = 1\right] \left(w(x, y) \cdot \ell_{reg}\left(\mathbf{p}_i(x, y), \mathbf{p}_i^*(x, y)\right)\right)$$

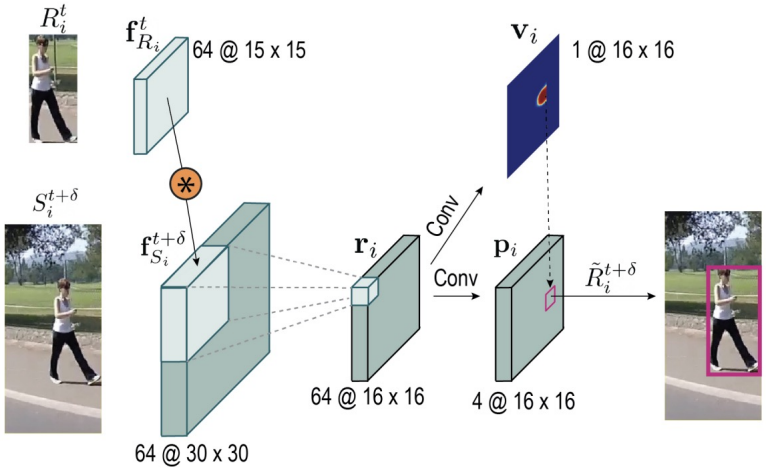

Fig. 6. *Network architecture of Emplicit Motion Model (EMM)* represents channel-wise cross correlation operator.

EMM improves upon the IMM design in two ways. First it uses the channel independent correlation operation to allow the network to explicitly learn a matching function between the same instance in sequential frames. Second, it enables a mechanism for finer-grained pixel-level supervision which is important to reduce the cases of falsely matching to distractors.

4 EXPERIMENT

4.1 MOT challenge

There are many challenges on MOT (e.g MOT15, MOT17, MOT20), we here introduce the MOT17 [3] for example in this proposal. MOT17 is a standardized benchmark for a fair evaluation of single camera multi-person tracking methods. MOT17 presented its first two data releases with about 35,000 frames of footage and almost 700,000 annotated pedestrians.
The first number indicates in which frame the object appears, while the second number identifies that object as belonging to a trajectory by assigning a unique ID (set to -1 in a detection file, as no ID is assigned yet). Each object can be assigned to only one trajectory. The next four numbers indicate the position of the bounding of the pedestrian in 2D image coordinates. The position is indicated by the top left corner as well as the width and height of the bounding box. A single number following denotes the detection confidence score.

4.1.1 Singapore Maritime Dataset. Few marine datasets exist in the research community because most applications are commercial or military. Singapore Maritime Dataset(SMD) [7] is a public dataset, using Canon 70D cameras around Singapore waters. SMD has 81 video files, including 240,842 target tags in 9 categories. All videos are acquired in high definition (1080x1920 pixels). Many vessels in the videos, such as buoys, speedboats, kayaks, and ships, have considerable variation in scale, making them very challenging for the detection algorithm. AS shown in Fig.7, SMD divides the dataset into parts, on-shore videos and on-board videos, which are acquired by camera placed on-shore on fixed platform and camera placed on-board a moving vessel, respectively.

Subdataset	Videos (Annotated)	Labeled Frames	Number of Labels
NIR	30 (23)	11,286	83,174
VIS on-board	11 (4)	2,400	3,173
VIS on-shore	40 (36)	17,967	154,495
Total	81 (63)	31,653	240,842

Fig. 7. *Dataset composition*

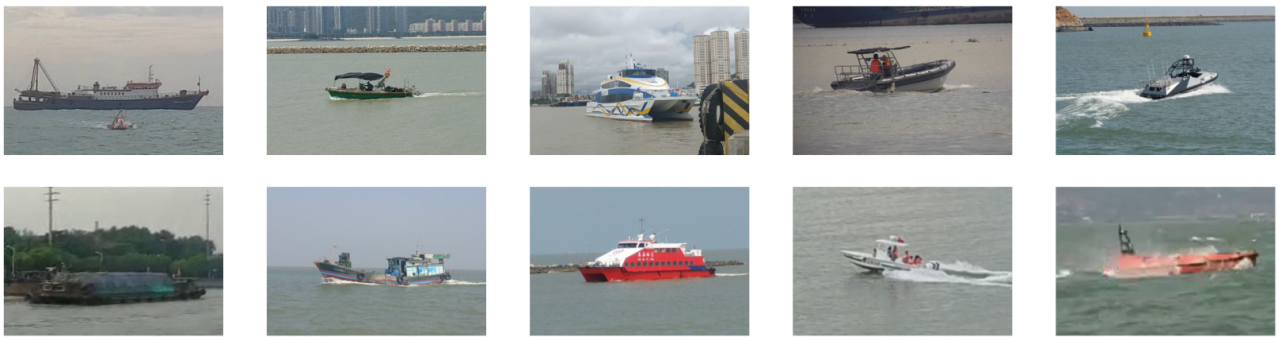

Fig. 8. *The ships in SMD*

The videos are acquired at various locations and routes and thus do not necessarily capture the same scene. The third part is Near Infra red (NIR) videos which is also captured using another Canon 70D camera with hot mirror removed and Mid-Opt BP800 Near-IR Bandpass filter.

Table 1. *MOT17 [3] challenge's data format for the input and output files, both for detection (DET) and annotation/ground truth (GT) files.*

Position	Name	Description
1	Frame number	Indicate at which frame the object is present
2	Identity number	Each pedestrian trajectory is identified by a unique ID (-1 for detections)
3	Bounding box left	Coordinate of the top-left corner of the pedestrian bounding box
4	Bounding box top	Coordinate of the top-left corner of the pedestrian bounding box
5	Bounding box width	Width in pixels of the pedestrian bounding box
6	Bounding box height	Height in pixels of the pedestrian bounding box
7	Confidence score	DET: Indicates how confident the detector is that this instance is a pedestrian.
8	Class	GT: Indicates the type of object annotated
9	Visibility	GT: a visibility ratio between 0 and 1 that indicates the visibility of object.

Table 2. *Some comparisons between popular MOT methods implemented for MST.*

Method/Metric	MOTA(↑)	MOTP(↑)	ID-F1(↓)
SORT[2]	31.4	0.219	55.0
DeepSORT[9]	31.6	0.224	54.4
Motdt[6]	13.2	NaN	28.4
ByteTrack[12]	33.8	0.225	57.8
SiamMOT[8]	47.7	0.244	68.7

Table 3. *A comparison with the latest released method RoDAN[11] specialized for MST.*

Method/Metric	ID-F1(↑)	ML(↓)	FP(↓)	FN(↓)	ID-s(↓)	MOTA(↑)	MOTP(↑)
RoDAN[11]	55.7	30	2977	17158	59	46.2	55.3
SiamMOT[8]	68.7	28	6876	15158	217	47.7	24.4

4.1.2 MOT17. The data format of SMD is based on Matlab files, which we need to convert to MOT17 format.MOT17, known as Multiple Object Tracking 17, is a dataset that can be used for multi-object tracking. The table 1 describes in detail the format of MOT17. Each line represents one object instance and contains 9 values . The last three numbers indicate the 3D position in real-world coordinates of the pedestrian, which can be left at -1 when it comes to 2D.

In our work, videos in SMD will be converted to JPEG format and named sequentially with a 6-digit file name (e.g. 000001. jpg). Detection and annotation files will be written into simple comma-separated value (CSV) files.

4.2 Evaluation metrics

we utilise the evaluation metrics defined in [5], along with the standard MOT metrics [1] 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.

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

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

4.3 Comparisons with state-of-the-art methods

4.3.1 Quantitative analysis.

Fig. 9. *From top to bottom, there are four frames, frame100, frame120, frame140, frame160, from MVI0801VISOB[7] dataset. From left to right, they are SORT, DeepSORT, ByteTrack, and SiamMOT, respectively.*

Fig. 10. *A comparison with SiamMOT and ByteTrack, ByteTrack is one of the most popular MOT method in this year.*

4.3.2 Qualitative analysis. We demonstrate a comparison between different methods as Fig. 9 shown. Four consecutive frames were selected from the videos at twenty frame intervals for the scenario where the camera was shaking violently. From left to right, they are SORT[2], DeepSORT[9], ByteTrack[12], and SiamMOT[8] respectively. From Fig. 9 we can see, the tracking of SORT is unstable, the tracking bounding box is not continuously; Deepsort also has a problem of ID-Switch facing to shaking scenario. The ByteTrack also has an error detection; And the SiamMOT is very stable and accurate.

4.4 Limitation

SiamMOT also has some limitations. As shown in Fig. 10 We can see that SiamMOT still should deal with the problem of small object detection performance. Specifically, the ByteTrack utilizes a yolov5 detector, but the SiamMOT still depends on Faster R-CNN.

5 CONCLUSION

In this paper, we have extensively explored the application of MOT in MST. Starting from traditional computer vision based methods to modern deep learning based methods, this paper not only emphasizes the importance of traditional tracking architecture, but also explores the application of deep neural network. It is a famous and effective mechanism to detect objects before tracking, this paper also demonstrate tracking is a combination task of detection and association. Nowadays, the performance of detection is soaring up rapidly, some detectors like Yolo series and transformer-based detectors have dominated the target detection field. However, the association related researches are still stagnant. The SiamMOT we implemented has shown great strength, which pay more attention to the association part to enhance the capability of motion modeling. As we discussed in Part 4, the deep learning based motion modeling surpasses other traditional and re-id-based model by a large margin, which also indicates our future research direction.

REFERENCES

[1] Keni Bernardin and Rainer Stiefelhagen. 2008. Evaluating multiple object tracking performance: the clear mot metrics. *EURASIP Journal on Image and Video Processing* 2008 (2008), 1–10.

[2] Alex Bewley, Zongyuan Ge, Lionel Ott, Fabio Ramos, and Ben Upcroft. 2016. Simple online and realtime tracking. In *2016 IEEE International Conference on Image Processing (ICIP)*. 3464–3468. https://doi.org/10.1109/ICIP.2016.7533003

[3] Patrick Dendorfer, Aljosa Osep, Anton Milan, Konrad Schindler, Daniel Cremers, Ian Reid, Stefan Roth, and Laura Leal-Taixé. 2021. Motchallenge: A benchmark for single-camera multiple target tracking. *International Journal of Computer Vision* 129, 4 (2021), 845–881.

[4] Gereon Hinz, Guang Chen, Muhammad Aafaque, Florian Röhrbein, Jörg Conradt, Zhenshan Bing, Zhongnan Qu, Walter Stechele, and Alois Knoll. 2017. Online multi-object tracking-by-clustering for intelligent transportation system with neuromorphic vision sensor. In *Joint German/Austrian Conference on Artificial Intelligence (Künstliche Intelligenz)*. Springer, 142–154.

[5] Yuan Li, Chang Huang, and Ram Nevatia. 2009. Learning to associate: Hybridboosted multi-target tracker for crowded scene. In *2009 IEEE conference on computer vision and pattern recognition*. IEEE, 2953–2960.

[6] Chen Long, Ai Haizhou, Zhuang Zijie, and Shang Chong. 2018. Real-time Multiple People Tracking with Deeply Learned Candidate Selection and Person Re-identification. In *ICME*.

[7] Dilip K Prasad, Deepu Rajan, Lily Rachmawati, Eshan Rajabally, and Chai Quek. 2017. Video processing from electro-optical sensors for object detection and tracking in a maritime environment: A survey. *IEEE Transactions on Intelligent Transportation Systems* 18, 8 (2017), 1993–2016.

[8] Bing Shuai, Andrew Berneshawi, Xinyu Li, Davide Modolo, and Joseph Tighe. 2021. Siammot: Siamese multi-object tracking. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 12372–12382.

[9] Nicolai Wojke, Alex Bewley, and Dietrich Paulus. 2017. Simple Online and Realtime Tracking with a Deep Association Metric. In *2017 IEEE International Conference on Image Processing (ICIP)*. IEEE, 3645–3649. https://doi.org/10.1109/ICIP.2017.8296962

[10] Fangao Zeng, Bin Dong, Yuang Zhang, Tiancai Wang, Xiangyu Zhang, and Yichen Wei. 2022. Motr: End-to-end multiple-object tracking with transformer. In *European Conference on Computer Vision*. Springer, 659–675.

[11] Wen Zhang, Xujie He, Wanyi Li, Zhi Zhang, Yongkang Luo, Li Su, and Peng Wang. 2021. A robust deep affinity network for multiple ship tracking. *IEEE Transactions on Instrumentation and Measurement* 70 (2021), 1–20.

[12] Yifu Zhang, Peize Sun, Yi Jiang, Dongdong Yu, Fucheng Weng, Zehuan Yuan, Ping Luo, Wenyu Liu, and Xinggang Wang. 2022. Bytetrack: Multi-object tracking by associating every detection box. In *European Conference on Computer Vision*. Springer, 1–21.