PMTrack: Multi-object Tracking with Motion-Aware

*Xu Guo, *Yujin Zheng, and Dingwen Wang

School of Computer Science, Wuhan University, Wuhan 430072, Hubei, China
wangdw@whu.edu.cn

Abstract. Tracking-by-detection typically involves associating detection boxes across frames in a video sequence. A common approach is to use Kalman filter for prediction and matching with detection boxes based on IoU. However, the Kalman filter is a linear prediction method, which, in scenarios involving camera motion or nonlinear object motion, will result in issues like ID switching or tracking loss. To address the problem, we propose a method that leverages phase correlation to calculates the translational relationship between adjacent frames, maping target positions into the current frame's coordinate system. This positional correction effectively compensates for the shifts caused by camera movement, significantly reducing ID switches. Furthermore, our method distinguishes between the motion and stationary states of trajectories, thereby enhancing tracking stability and accuracy. Our experimental results demonstrate that the proposed approach attains real-time efficiency and excels in scenes with camera motion. It achieves an MOTA of 80.17%, IDF1 of 78.93%, and HOTA of 64.04% on the MOT17 test sets, surpassing mainstream works in terms of multiple performance indicators.

Keywords: Mutli-object tracking · Tracking-by-detection · Kalman filter · phase correlation.

1 Introduction

Multi-object tracking (MOT) is a fundamental problem in the realm of computer vision, aiming to accurately detect and track multiple objects simultaneously in dynamic environments. MOT boasts extensive applications across diverse domains such as autonomous driving, smart transportation systems, and video analytics. The task involves the continuous monitoring and localization of multiple objects within a video stream, enabling crucial functionalities like object interaction analysis, behavior understanding, and scene understanding.

Tracking-by-detection has emerged as the prevailing paradigm in the contemporary field of multi-object tracking. The process entails initially detecting objects within each frame to ascertain their spatial positions. Subsequently, these detections are linked across consecutive frames to form coherent trajectories, providing a comprehensive understanding of object movements and interactions over time.

*These authors contributed to the work equally and should be regarded as co-first authors.

Within tracking-by-detection methodologies, Kalman filter emerges as a prevalent choice for state estimation and trajectory prediction. Kalman filter is adept at forecasting the positions of tracked objects in the succeeding frames by incorporating information from previous observations and predicting the evolution of object states over time. These predicted positions are then matched with detection boxes from the current frame using techniques like Intersection over Union (IoU) to establish associations between predicted trajectories and detected objects.

However, despite the efficacy of Kalman filter, challenges persist, particularly in complex scenarios characterized by camera motion. In such situations, the predicted positions of tracking boxes may deviate significantly from the corresponding detection boxes due to camera motion-induced distortions. Consequently, the overlap between predicted and detected boxes diminishes, leading to association failures and consequent degradation in tracking performance.

To address the challenges posed by camera motion in multi-object tracking, the method we propose leverages phase correlation to compute the translational relationship between consecutive frames in a video sequence. By accurately determining the translation between frames, we can effectively compensate for the displacement induced by camera motion. This information is then utilized to refine the state vector of the Kalman filter. This correction mechanism significantly reduces instances of tracking loss and improves the overall robustness of the tracking system in dynamic environments, ensuring more accurate trajectory predictions. Moreover, we can further distinguish stationary and moving tracklets, allowing for tailored association strategies based on the motion states of objects. By adjusting the association threshold for stationary trajectories, we prioritize the matching of these objects with detection boxes, thereby enhancing the continuity and reliability of tracking results. The main contributions of this work can be summarized as follows:

- We propose a camera motion compensation module named Translation-based Prediction Modification(TPM), which modifies the predicted bounding box positions based on the translational relationship between adjacent frames. This effectively alleviates association failures caused by camera motion, resulting in a significant reduction in the number of ID switches.
- We propose a new association method named Tracklets State Prior(TSP), which distinguishes the motion states of tracklets based on prior information. For static targets, we prioritize association and narrow down the range of participating detection boxes, aiming to achieve more accurate matching.

2 Related Work

2.1 Tracking by detection method

The basic idea of tracking-by-detection is to detect and associate objects. Firstly, each frame's objects are detected using a detection algorithm. Then, a matching algorithm is used to associate objects in adjacent frames, forming tracking

trajectories for each object. Hence, the accuracy and reliability of the detector significantly influence the overall tracking quality. SORT[3] employs Kalman filter to predict the state of each detection box in the current frame, thereby obtains predicted tracking boxes through state estimation. It then uses the Hungarian algorithm based on IoU to match the tracking boxes and detection boxes, thereby updating the Kalman filter parameters. deepSORT[26] enhances SORT by incorporating appearance information, which improves tracking accuracy by leveraging the visual features of objects. Additionally, it employs a cascaded matching strategy for robust association. GGDA[27] designs a graph network for tracklets grouping, using min-cost flow for intra-group association and hypothesis proposals with pruning for inter-group association to address long-term occlusion and reduce false positives. ByteTrack[31] introduces a novel association method that preserves nearly all detection boxes for matching, thereby mitigating the issue of discarding low-score boxes caused by occlusion or motion blur. OCSORT[5] focuses on observations, smoothing the parameters of the trajectory loss state to reduce noise accumulation during prediction. TrackFlow[13] introduces a depth estimation network to infer the distance between targets and the camera, incorporating depth information into the association process. MotionTrack[17] devises an interaction module to replace Kalman filtering for motion prediction, alongside a refine module to stitch fragmented tracklets.

Traditional methods relying on Kalman filter for state prediction encounter limitations due to the linear modeling assumption. In real complex scenarios, camera motion introduces non-linear object motion, leading to mismatching between detection boxes and tracking trajectories. We employ phase correlation method to adjust tracking trajectories, thereby compensating for camera motion and ensuring accurate matching between the tracking trajectories and detection boxes.

2.2 Joint detection and tracking method

The joint detection and tracking framework employs a shared neural network to simultaneously learn detection and tracking tasks. By employing a multi-task learning paradigm, the framework shares feature learning network parameters and defines loss functions, facilitating interaction and mutual promotion between detection and tracking. JDE[25] incorporates the feature extraction network into a unified single-stage object detection model, enabling the network to output results for both tasks and proposes a new association mechanism. FairMOT[32] employs two parallel heads that share extracted features, used separately for detection and re-identification tasks. CenterTrack[33], based on CenterNet[34], incorporates input from the previous frame and introduces an additional branch to predict the displacement of targets between consecutive frames, employing a greedy algorithm for association. These approaches showcase the advancement in joint detection and tracking by leveraging shared feature learning and innovative association mechanisms within neural network architectures. FineTrack[18] comprehensively describes appearance from both global and local perspectives, enhancing feature consistency and discrimination.

2.3 Transformer-based tracking method

With the widespread use of transformers[23] in the field of computer vision, their powerful inter-frame propagation capability has also found application in the domain of multi-object tracking. TransTrack[21] utilizes an attention-based query-key mechanism to decouple MOT into two sub-tasks, namely detection and association. Similarly, TrackFormer[14] jointly performs tracking and detection using a single decoder network. Additionally, MOTR[30], extending upon DETR[7], introduces a "tracking query" mechanism to represent and track objects across the entirety of a video sequence in an end-to-end fashion. MEMOTR[9] injects long-term memory into the track query, enhancing the utilization of temporal information.

3 Method

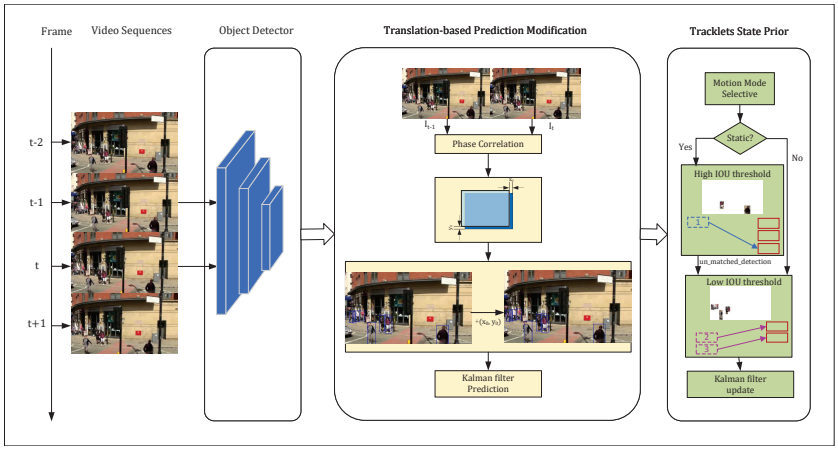


Fig. 1: The pipeline of our proposed PMTrack.

The method described in this paper, as depicted in Fig. 1, employs phase correlation to compute the translation relationship between consecutive frames in a video sequence. The offset is utilized to adjust the state vector of the Kalman filter. Both detection results and trajectory predictions are transformed into the coordinate system of the current frame. Subsequently, association is performed separately for moving and stationary trajectories using the approach proposed in Byte[31]. By mitigating matching losses and ID switches induced by camera motion, this approach significantly improves tracking performance.

3.1 Translation-based Prediction Modification

In tracking-by-detection, matching between detection boxes and Kalman filter-predicted tracking boxes is typically performed by calculating the IOU score. The effectiveness of tracking heavily depends on the degree of overlap between detection and tracking boxes, as it directly influences the accuracy of association. Abrupt camera movements can lead to substantial displacements of detection boxes relative to their positions in the previous frame, leading to some detection boxes failing to associate with predicted tracking boxes. This challenge is especially pronounced for smaller targets, resulting in track losses or increased ID switches. Botsort[1] employs the GMC algorithm implemented in OpenCV to estimate motion between image backgrounds. This process involves extracting feature points from consecutive frames and utilizing sparse optical flow for feature tracking. The algorithm computes the affine transformation matrix between adjacent frames to transform the coordinates of predicted bounding boxes into the current frame's coordinate system. Although this approach enhances performance in scenarios with moving cameras, it imposes significant computational overhead, leading to a notable reduction in processing speed and failing to satisfy real-time constraints for multi-object tracking.

Adjacent frames in a video sequence exhibit a certain translational relationship due to camera motion. Considering the translational invariance property of the Fourier transform of images, which is reflected in the phase, we integrate the phase correlation method used in image registration into the trajectory association process of multi-object tracking. It calculates the translational transformation between images and transforms the prediction process of tracking trajectories from the coordinate system of the previous frame to that of the current frame.

Phase correlation is an algorithm used to address registration problems between two images with translational relationships. The foundation of phase correlation lies in the Fourier transform of images. For two images $f_1(x, y)$ and $f_2(x, y)$, it satisfies:

$$f_2(x, y) = f_1(x + x_0, y + y_0) \tag{1}$$

According to the properties of Fourier transform, when reflected in the frequency domain, we have:

$$F_2(u, v) = F_1(u, v) * e^{-j2\pi(ux_0+vy_0)} \tag{2}$$

Where F_1 and F_2 are the Fourier transforms of $f_1(x, y)$ and $f_2(x, y)$ respectively. Computing their cross-power spectrum yields:

$$H_{(u,v)} = \frac{F_1^*(u, v) F_2(u, v)}{|F_1^*(u, v) F_2(u, v)|} = e^{-j2\pi(ux_0+vy_0)} \tag{3}$$

Where $F_1^*(u, v)$ is the complex conjugate of $F_1(u, v)$. Applying the inverse Fourier transform to Eq. 3 results in an impulse function, with the coordinates of its peak denoted as (x_0, y_0), representing the translational relationship between images.

$$\mathbf{x} = \left[x + x_0, y + y_0, a, h, \dot{x}, \dot{y}, \dot{a}, \dot{h}\right] \tag{4}$$

Adjusting the state vector used by the Kalman filter for each tracking trajectory, as shown in Eq. 4, allows for transforming the tracking trajectory into the coordinate system of the current frame. Subsequently, subsequent operations such as Kalman filter prediction, association, and update can be performed.

3.2 Tracklets State Prior

In conventional tracking-by-detection architectures, all tracking trajectories are usually associated with detection boxes using a uniform IoU threshold. However, in typical scenes, a mixture of static and moving targets is present. The positional variation of static targets in images is notably less pronounced compared to that of moving targets. It is more appropriate to use distinct IoU thresholds for matching static and moving targets separately, as Fig. 2.

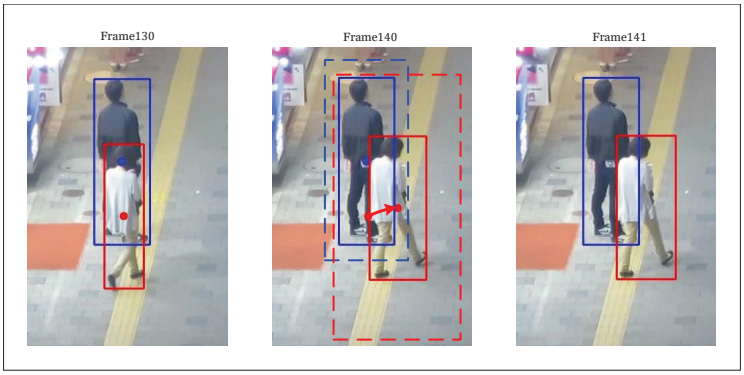


Fig. 2: Illustration of the Tracklets State Prior. The blue box has been static for the past ten frames and is suitable for a small association range, such as the blue dotted box. The red box is in motion and is suitable for a large association range, such as the red dotted box. The images are from the MOT17-04 sequence.

We utilize prior information from tracklets to determine their motion states. Tracklets exhibiting high overlap among associated detection boxes over a consecutive number of frames are classified as stationary, constituting static tracklets, while others are categorized as motional tracklets. The association process follows Byte[31] methodology, wherein the tracklets are firstly matched with the high-scoring detection frame, and subsequently re-matched with the low-scoring detection frame if unmatched in the first round. Static tracklets are prioritized for associating, employing a higher IOU threshold to narrow down the range of matching detection boxes, thereby achieving more accurate matching. Motional tracklets, on the other hand, engage in association with a larger range of unmatched detection boxes.

3.3 Algorithm Description

The pseudo-code of PMTrack is shown in Algorithm.1. The input of PMTrack is a video sequence V, along with an object detector Det and a detection score threshold τ. We also set high IOU threshold $threshold_{high}$, low IOU threshold $threshold_{low}$, static tracklet threshold α and static consecutive frame number n. This $threshold_{high}$ is used as the threshold of matching between static tracklets and detections while $threshold_{low}$ is used as the threshold of matching between move tracklets and detections. α and n are used to judge the motion state of tracklets. The output of PMTrack is the tracks T of the video and each track contains the bounding box and identity of the object in each frame.

Line 7 to 12 in Algorithm.1 represent our TPM module. This module involves computing the translation between adjacent frames using phase correlation and adjusting the state vectors of the tracks accordingly. This modification effectively reduces the number of association failures caused by camera motion, ensuring more accurate predictions of object positions.

Line 15 to 34 in Algorithm.1 show the TSP method. The tracks are split into static and moving categories based on their state and then associate with detections using different IOU threshold. The Byte method in Algorithm.1 is come from ByteTrack[31], which means associating with both high-score and low-score boxes.

4 Experiments

4.1 Experimental Settings

Datasets. The datasets used in this study include MOT17[15] and Kitti[11]. MOT17 is a common dataset in the field of multi-object tracking, primarily used for pedestrian tracking. It consists of multiple video sequences captured by static or dynamic cameras, which are divided into training and testing sets. In the final evaluation of the MOT challenge, only the boxes marked as pedestrians are used. Therefore, our tests on the MOT17 dataset are only for the category of pedestrians. Kitti is a multi-object tracking dataset collected using cameras installed on moving vehicles. It provides annotations for various classes, including pedestrians, cars, cyclists, and trucks, among osthers. It comprises 21 training video sequences and 29 testing video sequences.

Implementation details. For fair comparison, we utilize the same detection results as ByteTrack in our experiments on MOT17. ByteTrack employs YOLOX[10] as the detector and has trained a well-performing detection model on MOT17. We utilize the provided bytetrack_ablation.pth.tar from the ByteTrack open-source project for ablation study on MOT17, which is trained on Crowdedhuman[20] and the first half of MOT17 training sets. The detection model used on the MOT17 test set is bytetrack_x_mot17.pth.tar from ByteTrack. Our approach achieves a processing speed of 20.2 FPS on the MOT17

Algorithm 1 Pseudo-code of PMTrack

Input: A video sequence V; object detector Det; detection score threshold τ; high IOU threshold $threshold_{high}$; low IOU threshold $threshold_{low}$; static tracklet threshold α; static consecutive frame number n

Output: Tracks $\mathcal{T}$ of the video

Initialization: $\mathcal{T} \leftarrow \emptyset$
for frame f_k in V **do**
  $\mathcal{D}_k \leftarrow Det(f_k)$
  $\mathcal{D}_{high} \leftarrow \emptyset$, $\mathcal{D}_{low} \leftarrow \emptyset$, $\mathcal{T}_{static} \leftarrow \emptyset$, $\mathcal{T}_{move} \leftarrow \emptyset$
  $\mathcal{D}_{high}$, $\mathcal{D}_{low} \leftarrow$ split D_k with threshold τ
  /* Translation-based Prediction Modification */
  if $k > 1$ **then**
    $(x_0, y_0) \leftarrow$ phaseCorrelation(f_k, f_{k-1})
    for t *in* $\mathcal{T}$ **do**
      t.kalmanfilter.stateVector $\leftarrow$ t.kalmanfilter.stateVector+(x_0, y_0)
    end for
  end if
  KalmanFilter.predict($\mathcal{T}$)
  /* Tracklets State Prior: association */
  $\mathcal{T}_{static}$, $\mathcal{T}_{move} \leftarrow$ split $\mathcal{T}$ with $t.state$
  t.box=d, association T_{static} with D_{high} and D_{low} using Byte, IOU_threshold=$threshold_{high}$
  $\mathcal{D}_{remian\_high} \leftarrow$ remaining object boxes from $\mathcal{D}_{high}$
  $\mathcal{D}_{remian\_low} \leftarrow$ remaining object boxes from $\mathcal{D}_{low}$
  t.box=d, association T_{move} with D_{remian_high} and D_{remian_low} using Byte, IOU_threshold=$threshold_{low}$
  $\mathcal{D}_{remian} \leftarrow$ remaining object boxes from $\mathcal{D}_{remian\_high}$
  $\mathcal{T}_{remian} \leftarrow$ remaining tracks from $\mathcal{T}_{static}$ and T_{move}
  /* Tracklets State Prior: Tracklets State setting */
  $\mathcal{T} \leftarrow (\mathcal{T}_{static} \cup \mathcal{T}_{move})/\mathcal{T}_{remian}$
  for t *in* $\mathcal{T}$ **do**
    if $IOU(t.box, t.last\_box) > \alpha$ **then**
      $t.consecutive\_frame \leftarrow t.consecutive\_frame + 1$
      if $t.consecutive\_frame > n$ **then**
        $t.state \leftarrow static$
      end if
    else
      $t.state \leftarrow move$, $t.consecutive\_frame \leftarrow 0$
    end if
    $t.last\_box \leftarrow t.box$
  end for
  for d *in* $\mathcal{D}_{remian}$ **do**
    $\mathcal{T} \leftarrow \mathcal{T} \cup \{d\}$
  end for
end for
return $\mathcal{T}$

test set. For ablation study on Kitti, the detection model employs an image size of 1280x384 for both training and testing. We train the model for 80 epochs on the first half of the Kitti object tracking dataset and use the latter half as the validation set. We define stationary trajectories as those formed by matching detection boxes between adjacent frames with an IOU exceeding 0.95 and persisting for at least 10 consecutive frames. The IOU threshold for matching stationary trajectories with detection boxes is set to 0.7.

Evaluation metrics. The study employs commonly used metrics in the multi-object tracking domain, including the CLEAR[2] metrics ($MOTA$, FN, FP, $IDSW$), $IDF1$, and $HOTA$[12], to assess tracking performance.

$MOTA_t$ is defined as follows:

$$MOTA = 1 - \frac{\sum_t (FP_t + FN_t + IDSW_t)}{\sum_t GT_t} \tag{5}$$

Where GT_t, FP_t, FN_t, and $IDSW_t$ represent the number of ground-truth bounding boxes, false positives, false negatives, and identity switches, respectively, at frame t. $MOTA$ quantifies the proportion of correctly predicted samples without missed detections, false alarms, and identity switches among all annotated samples, measuring the tracker's performance in detecting targets and maintaining trajectories.

$IDF1$ is defined as follows:

$$IDF1 = \frac{2IDTP}{2IDTP + IDFP + IDFN} \tag{6}$$

$IDF1$ treats each ID as a separate class, considering both ID accuracy and ID recall, making it more sensitive to ID information within trajectories than $MOTA$.

$HOTA$ is defined as follows:

$$HOTA = \sqrt{\frac{\sum_{c \in \{TP\}} \mathcal{A}(c)}{|TP| + |FN| + |FP|}} \tag{7}$$

Where $\mathcal{A}(c)$ is the association accuracy score, measuring the alignment degree of true trajectories.

4.2 Ablation Study

The ablation study takes ByteTrack as the baseline. The second half of MOT17 training set video sequences is used as the validation set. All parameters remain consistent across the ablation study, and the experimental results are shown in Table 1.

The phase correlation method is primarily used to correct the positions of tracking trajectories in scenes with camera motion, aligning them with detection boxes. Therefore, ablation studies are conducted on MOT17 training sets based

Table 1: Ablation study on the MOT17 validation set.

Method	TPM	TSP	MOTA↑	IDF1↑	IDs↓
Baseline (ByteTrack)			76.6	79.4	159
Baseline + column 1	✓		76.9	80.4	126
Baseline + column 1-2	✓	✓	77.4	80.8	124

on whether the camera moves frequently. MOT17-05, MOT17-10, MOT17-11, and MOT17-13 form the MOT17_move dataset, representing camera motion scenes, while MOT17-02, MOT17-04, and MOT17-09 form the MOT17_static dataset, representing static camera scenes. The detection model used is bytetrack_ablation.pth.tar, and the experiments on these two datasets yield results as shown in Table 2.

Table 2: Ablation study on the MOT17 validation set for different scenarios.

	MOT17_move			MOT17_static		
Method	MOTA↑	IDF1↑	IDs↓	MOTA↑	IDF1↑	IDs↓
Baseline (ByteTrack)	78.2	70.5	287	85.7	82.7	150
Baseline + TPM	79.4	75.8	157	85.7	83.1	149

Table 2 presents a comparison of tracking performance between camera motion and stationary scenes in MOT17. It is evident that the tracking performance in stationary scenes significantly outperforms that in camera motion scenes, indicating that camera motion indeed poses challenges to multi-object tracking. Upon employing motion compensation using phase correlation, notable improvements are observed in tracking performance in camera motion scenes, particularly in reducing ID switches. However, the improvement in tracking performance in stationary scenes is marginal.

The length of historical tracklets is a critical hyperparameter in the TSP module. As shown in Fig.3, we designed an ablation study for this parameter, and the results demonstrate that TSP exhibits robustness to variations in this parameter.

Table 3: Ablation study on the Kitti validation set for the Car category.

Method	TPM	TSP	MOTA↑	IDF1↑	HOTA↑	IDs↓
Baseline (ByteTrack)			80.091	87.029	71.195	57
Baseline + column 1	✓		80.292	87.494	71.747	40
Baseline + column 1-2	✓	✓	80.62	87.658	71.788	42

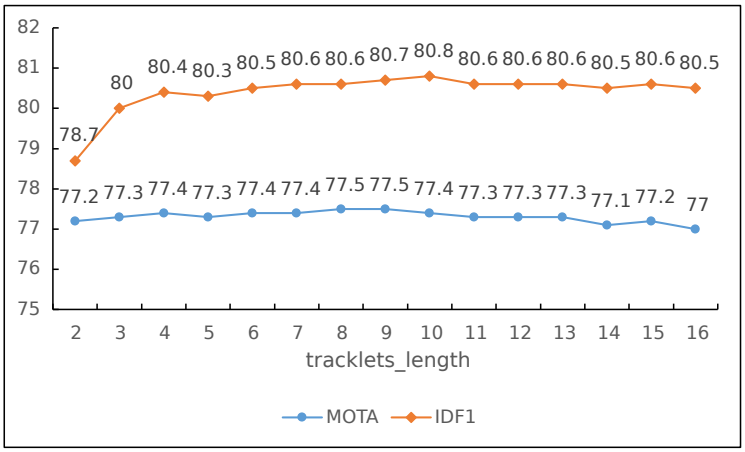


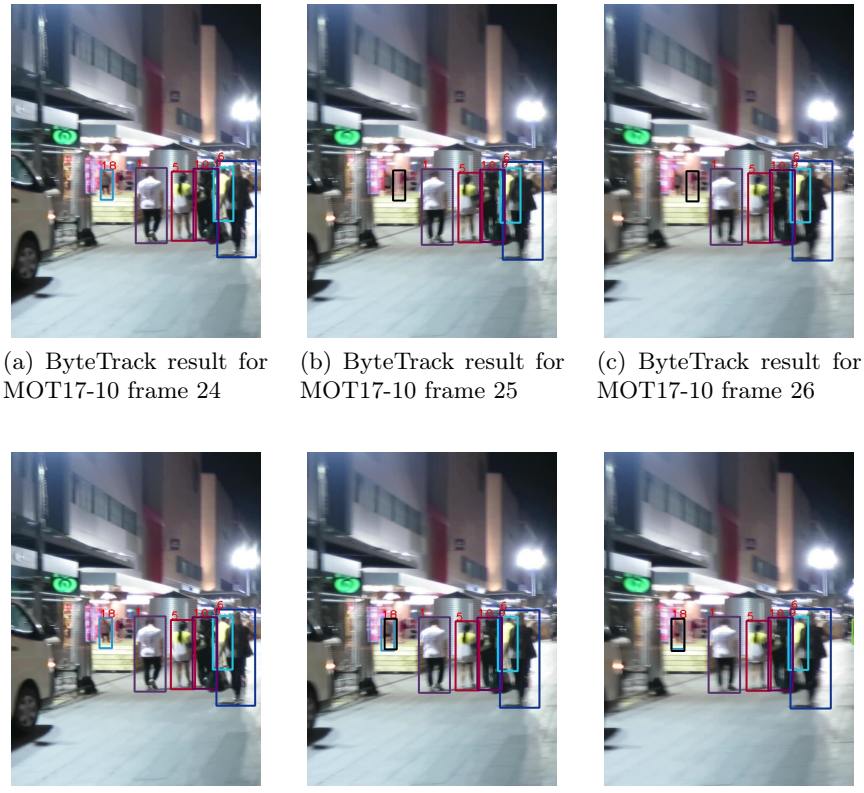
Fig. 3: Ablation study on tracklet length for TSP.

For the ablation study on the Kitti object tracking dataset, the first half of the training set is used for training, and the second half is used to construct the validation set.

Ablation study on the MOT17 and Kitti validation sets, as presented in Tables 1 and 3, assess the contributions of the proposed modules in PMTrack. The experimental results validate the effectiveness of the modules proposed in PMTrack. In IoU-based multi-object tracking association methods, camera motion often results in significant changes in target positions compared to the previous frame, thereby increasing the difficulty of correct association and leading to numerous ID switches. To address this challenge, we propose an association method based on phase correlation. This method calculates the translational relationship between adjacent frames, mapping the target positions from the previous frame and the current frame into the coordinate system of the current frame. By effectively compensating for the positional shifts caused by camera motion through this positional correction mechanism, we can significantly reduce the number of ID switches. However, Table 2 results indicate that phase correlation provides substantial assistance only on datasets with camera motion scenes. Furthermore, the method for distinguishing between motion and stationary trajectories enhances the precision of association, yielding higher tracking metrics.

4.3 Visualization results

We show some visualization results of difficult cases which PMTrack is able to handle in Fig. 4. From frame 24 to frame 26, the image scene undergoes a leftward translation. In Fig. 4b and Fig. 4c, the predicted box by the Kalman filter is positioned to the right of the target, failing to be associated via IoU, resulting in the loss of the 18th tracking box. In Fig. 4e and Fig. 4f, the translational



(a) ByteTrack result for MOT17-10 frame 24

(b) ByteTrack result for MOT17-10 frame 25

(c) ByteTrack result for MOT17-10 frame 26

(d) PMTrack result for MOT17-10 frame 24

(e) PMTrack result for MOT17-10 frame 25

(f) PMTrack result for MOT17-10 frame 26

Fig. 4: Visualization of the tracking result, where the black boxes in (b) ,(c)and (e),(f) represent the predicted bounding box of the 18th tracklet. The images are from the MOT17-10 sequence.

relationship between adjacent frames is computed, and the predicted box is also shifted leftward, successfully associated with the detection box.

4.4 Benchmarks Evaluation

In Table 4, we present the tracking performance of PMTrack using private detections on MOT17-test. For fair comparison, we utilize the same detection results as ByteTrack[31] and employ ByteTrack's linear interpolation method for post-processing the tracking results. Compared to other algorithms listed in the table, PMTrack demonstrates superior performance, particularly with a significant decrease in ID switches. Although PMTrack's MOTA performance is not as high as ByteTrack, ByteTrack's 80+ MOTA requires careful tuning of test image size

and detection thresholds. We used ByteTrack's default parameters without additional tuning, and simply using the open-source ByteTrack code does not reach the MOTA scores reported in our work.

Table 4: Results on MOT17-test with the private detections.

tracker	**HOTA↑**	**MOTA↑**	**IDF1↑**	**FP**(10^4)**↓**	**FN**(10^4)**↓**	**IDs↓**	**AssA↑**	**AssR↑**
FairMOT[32]	59.3	73.7	72.3	2.75	11.7	3,303	58	63.6
TransCenter[28]	54.5	73.2	62.2	2.31	12.4	4,614	49.7	54.2
TransTrack[21]	54.1	75.2	63.5	5.02	8.64	3,603	47.9	57.1
GRTU[24]	62	74.9	75	3.2	10.8	1,812	62.1	65.8
QDTrack[16]	53.9	68.7	66.3	2.66	14.7	3,378	52.7	57.2
MOTR[30]	57.2	71.9	68.4	2.11	13.6	2,115	55.8	59.2
PermaTrack[22]	55.5	73.8	68.9	2.9	11.5	3,699	53.1	59.8
TransMOT[8]	61.7	76.7	75.1	3.62	9.32	2,346	59.9	66.5
GTR[35]	59.1	75.3	71.5	2.68	11	2,859	61.6	-
DST-Tracker[6]	60.1	75.2	72.3	2.42	11	2,729	62.1	-
MeMOT[4]	56.9	72.5	69	2.72	11.5	2,724	55.2	-
UniCorn[29]	61.7	77.2	75.5	5.01	**7.33**	5,379	-	-
ByteTrack[31]	63.1	**80.3**	77.3	2.55	8.37	2,196	62	68.2
OC-SORT[5]	63.2	78	77.5	**1.51**	10.8	1,950	63.2	67.5
MEMOTR[9]	58.8	72.8	71.5	2.65	12.5	1902	58.4	63.0
GHOST[19]	62.8	78.7	77.1	-	-	2325	-	-
PMTrack	**64.04**	80.17	**78.93**	2.66	8.39	**1,245**	**63.64**	**70.11**

In Table. 5, we compare our method with other SORT-based algorithms. As shown, our method performs similarly to BotSort, but achieves a speed that is 4 to 5 times faster.

Table 5: comparison on MOT17-test with the SORT-like.

tracker	**HOTA↑**	**MOTA↑**	**IDs↓**	**FPS↑**
ByteTrack[31]	63.1	80.3	2,196	29.6
BotSort[1]	65,0	80.5	1,212	4.5
OC-SORT[5]	63.2	78	1,950	29.0
PMTrack	64.04	80.17	1,245	20.2

5 Conclusion

We analyze the limitations of SORT-like tracking methods. Specifically, we identify the inability of Kalman filter to predict non-linear motion as a key limitation. As a consequence, predicted tracking trajectories fail to associate with detection boxes in scenes with camera motion. We propose PMTrack, leveraging phase correlation to determine the translational relationship between adjacent frames, allowing us to map target positions into the current frame's coordinate system.

This positional correction compensates for shifts caused by camera motion, leading to a significant reduction in ID switches. Moreover, distinguishing the motion states of tracking trajectories and applying different association thresholds further enhances the accuracy of the associations. PMTrack exhibits superior tracking performance in dynamic camera scenes while maintaining simplicity and real-time capability. Experimental results on multiple datasets validate the effectiveness of PMTrack for multi-object tracking in scenes with camera motion.

References

1. Aharon, N., Orfaig, R., Bobrovsky, B.Z.: Bot-sort: Robust associations multi-pedestrian tracking. arXiv preprint arXiv:2206.14651 (2022)
2. Bernardin, K., Stiefelhagen, R.: Evaluating multiple object tracking performance: the clear mot metrics. EURASIP Journal on Image and Video Processing **2008**, 1–10 (2008)
3. Bewley, A., Ge, Z., Ott, L., Ramos, F., Upcroft, B.: Simple online and realtime tracking. In: 2016 IEEE international conference on image processing (ICIP). pp. 3464–3468. IEEE (2016)
4. Cai, J., Xu, M., Li, W., Xiong, Y., Xia, W., Tu, Z., Soatto, S..: Memot: Multi-object tracking with memory. 2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) pp. 8080–8090 (2022)
5. Cao, J., Pang, J., Weng, X., Khirodkar, R., Kitani, K.: Observation-centric sort: Rethinking sort for robust multi-object tracking. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 9686–9696 (2023)
6. Cao, J., Wu, H., Kitani, K.: Track targets by dense spatio-temporal position encoding. arXiv preprint arXiv:2210.09455 (2022)
7. Carion, N., Massa, F., Synnaeve, G., Usunier, N., Kirillov, A., Zagoruyko, S.: End-to-end object detection with transformers. In: European conference on computer vision. pp. 213–229. Springer (2020)
8. Chu, P., Wang, J., You, Q., Ling, H., Liu, Z.: Transmot: Spatial-temporal graph transformer for multiple object tracking. 2023 IEEE/CVF Winter Conference on Applications of Computer Vision (WACV) pp. 4859–4869 (2021)
9. Gao, R., Wang, L.: Memotr: Long-term memory-augmented transformer for multi-object tracking. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 9901–9910 (2023)
10. Ge, Z., Liu, S., Wang, F., Li, Z., Sun, J.: Yolox: Exceeding yolo series in 2021. arXiv preprint arXiv:2107.08430 (2021)
11. Geiger, A., Lenz, P., Stiller, C., Urtasun, R.: Vision meets robotics: The kitti dataset. The International Journal of Robotics Research **32**(11), 1231–1237 (2013). https://doi.org/10.1177/0278364913491297
12. Luiten, J., Osep, A., Dendorfer, P., Torr, P., Geiger, A., Leal-Taixé, L., Leibe, B.: Hota: A higher order metric for evaluating multi-object tracking. International journal of computer vision **129**, 548–578 (2021)
13. Mancusi, G., Panariello, A., Porrello, A., Fabbri, M., Calderara, S., Cucchiara, R.: Trackflow: Multi-object tracking with normalizing flows. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 9531–9543 (2023)
14. Meinhardt, T., Kirillov, A., Leal-Taixe, L., Feichtenhofer, C.: Trackformer: Multi-object tracking with transformers. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 8844–8854 (2022)

15. Milan, A., Leal-Taixé, L., Reid, I., Roth, S., Schindler, K.: Mot16: A benchmark for multi-object tracking. arXiv preprint arXiv:1603.00831 (2016)
16. Pang, J., Qiu, L., Li, X., Chen, H., Li, Q., Darrell, T., Yu, F.: Quasi-dense similarity learning for multiple object tracking. 2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) pp. 164–173 (2020)
17. Qin, Z., Zhou, S., Wang, L., Duan, J., Hua, G., Tang, W.: Motiontrack: Learning robust short-term and long-term motions for multi-object tracking. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 17939–17948 (2023)
18. Ren, H., Han, S., Ding, H., Zhang, Z., Wang, H., Wang, F.: Focus on details: Online multi-object tracking with diverse fine-grained representation. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 11289–11298 (2023)
19. Seidenschwarz, J., Brasó, G., Serrano, V.C., Elezi, I., Leal-Taixé, L.: Simple cues lead to a strong multi-object tracker. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 13813–13823 (2023)
20. Shao, S., Zhao, Z., Li, B., Xiao, T., Yu, G., Zhang, X., Sun, J.: Crowdhuman: A benchmark for detecting human in a crowd. arXiv preprint arXiv:1805.00123 (2018)
21. Sun, P., Cao, J., Jiang, Y., Zhang, R., Xie, E., Yuan, Z., Wang, C., Luo, P.: Transtrack: Multiple object tracking with transformer. arXiv preprint arXiv:2012.15460 (2020)
22. Tokmakov, P., Li, J., Burgard, W., Gaidon, A.: Learning to track with object permanence. 2021 IEEE/CVF International Conference on Computer Vision (ICCV) pp. 10840–10849 (2021)
23. Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A.N., Kaiser, Ł., Polosukhin, I.: Attention is all you need. Advances in neural information processing systems **30** (2017)
24. Wang, S., Sheng, H., Zhang, Y., Wu, Y., Xiong, Z.: A general recurrent tracking framework without real data. 2021 IEEE/CVF International Conference on Computer Vision (ICCV) pp. 13199–13208 (2021)
25. Wang, Z., Zheng, L., Liu, Y., Li, Y., Wang, S.: Towards real-time multi-object tracking. In: European conference on computer vision. pp. 107–122. Springer (2020)
26. Wojke, N., Bewley, A., Paulus, D.: Simple online and realtime tracking with a deep association metric. In: 2017 IEEE international conference on image processing (ICIP). pp. 3645–3649. IEEE (2017)
27. Wu, Y., Sheng, H., Wang, S., Liu, Y., Xiong, Z., Ke, W.: Group guided data association for multiple object tracking. In: Proceedings of the Asian Conference on Computer Vision (ACCV). pp. 520–535 (December 2022)
28. Xu, Y., Ban, Y., Delorme, G., Gan, C., Rus, D., Alameda-Pineda, X.: Transcenter: Transformers with dense representations for multiple-object tracking. IEEE Transactions on Pattern Analysis and Machine Intelligence **45**, 7820–7835 (2021)
29. Yan, B., Jiang, Y., Sun, P., Wang, D., Yuan, Z., Luo, P., Lu, H.: Towards grand unification of object tracking. In: European Conference on Computer Vision (2022)
30. Zeng, F., Dong, B., Zhang, Y., Wang, T., Zhang, X., Wei, Y.: Motr: End-to-end multiple-object tracking with transformer. In: European Conference on Computer Vision. pp. 659–675. Springer (2022)
31. Zhang, Y., Sun, P., Jiang, Y., Yu, D., Weng, F., Yuan, Z., Luo, P., Liu, W., Wang, X.: Bytetrack: Multi-object tracking by associating every detection box. In: European conference on computer vision. pp. 1–21. Springer (2022)

32. Zhang, Y., Wang, C., Wang, X., Zeng, W., Liu, W.: Fairmot: On the fairness of detection and re-identification in multiple object tracking. International Journal of Computer Vision **129**, 3069–3087 (2021)
33. Zhou, X., Koltun, V., Krähenbühl, P.: Tracking objects as points. In: European conference on computer vision. pp. 474–490. Springer (2020)
34. Zhou, X., Wang, D., Krähenbühl, P.: Objects as points. arXiv preprint arXiv:1904.07850 (2019)
35. Zhou, X., Yin, T., Koltun, V., Krähenbühl, P.: Global tracking transformers. 2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) pp. 8761–8770 (2022)