IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received 3 April 2024, accepted 24 April 2024, date of publication 29 April 2024, date of current version 6 May 2024.

Digital Object Identifier 10.1109/ACCESS.2024.3394534

RESEARCH ARTICLE

An Effective Method for Semi-Online Multi-Object Tracking Refinement

MENGJIAO WANG[1], RUJIE LIU[1], SEPTIANA LINA[2], NARISHIGE ABE[2], AND SHIGEFUMI YAMADA[2]

[1]Fujitsu Research and Development Center Company Ltd., Beijing 100022, China
[2]Fujitsu Ltd., Tokyo 105-7123, Japan

Corresponding author: Mengjiao Wang (wangmengjiao@fujitsu.com)

ABSTRACT In multi-object tracking(MOT), identity(ID) switches (i.e., single tracklet containing different objects) are common. Here, we propose a semi-online tracking refinement method, where the ID switches are detected by monitoring the changes in appearance similarity within a short duration temporal window. When an ID switch occurs, frames containing different object will firstly enter the window, causing a large drop in appearance similarity. As the window moves forward, the ID switch frame will exit the window, causing an increase in appearance similarity since the window is about to be solely filled with the switched object. This 'drop-increase' pattern in appearance similarity within the moving temporal window can be used to identify the ID switch point. Frames containing switched object are then split from the original tracklet and attached to other tracklets based on the similarities among their multiple representative prototypes. Comparing to the baseline, our refinement method can significantly improve the IDF1 score on MOT17 and MOT20 in a real-time manner.

INDEX TERMS Multi-object tracking, tracking refinement, online processing.

I. INTRODUCTION

Identity(ID) switch is a common problem in multi-object tracking (MOT) due to occlusion in crowded scenes, and post-processing is usually used to alleviate this problem [1], [2], [3], [4], [5], [6], [7], [8], [9], [10]. The common procedure follows a two-step scheme: split and merge. In the split step, the tracklet is divided into several small pieces at the ID switch time point, aiming to make each piece correspond to only one person; while in the merge step, these pieces are connected to a longer trajectory according to their similarity.

For offline tracking refinement, where the entire frame sequences in the whole trajectory are used, clustering has been widely used as an effective approach to reduce ID switch errors [1], [11]. For online tracking which is performed in a frame-by-frame manner, to decide which tracklet the current frame (detected bounding box) belongs to, only the detections before this moment can be utilized. However, in offline method, the detections before and after this moment are both adopted in clustering, leading to better discrimination.

An example illustrating the difference between offline and online methods is shown in Figure 1 (1) and Figure 1 (3), where the object's appearance gradually changes in both 'cross' (ID switch happens) and 'occlude' (temporary occlusion) scenes at frame $t = 6$. But different consequences are obtained, that is, an ID switch occurs in former case but not in the latter one. For correct tracking, these two scenes should be distinguished. In the offline clustering method, since the entire trajectory is utilized, the two scenes can be simply separated by checking the number of clusters. In the case of 'cross', multiple clusters will be obtained since different IDs are contained; conversely, only one cluster is generated for the 'occlude' scene. Despite its effectiveness, the accuracy benefit of offline tracking refinement is gained at the cost of real-time characteristic, i.e., the result is obtained only after the whole trajectory is generated, making it unsuitable for the real-time tracking scenarios.

However, ID switch detection is difficult for online schemes, as the gradual change characteristics (e.g., frame

The associate editor coordinating the review of this manuscript and approving it for publication was Nagendra Prasad Pathak.

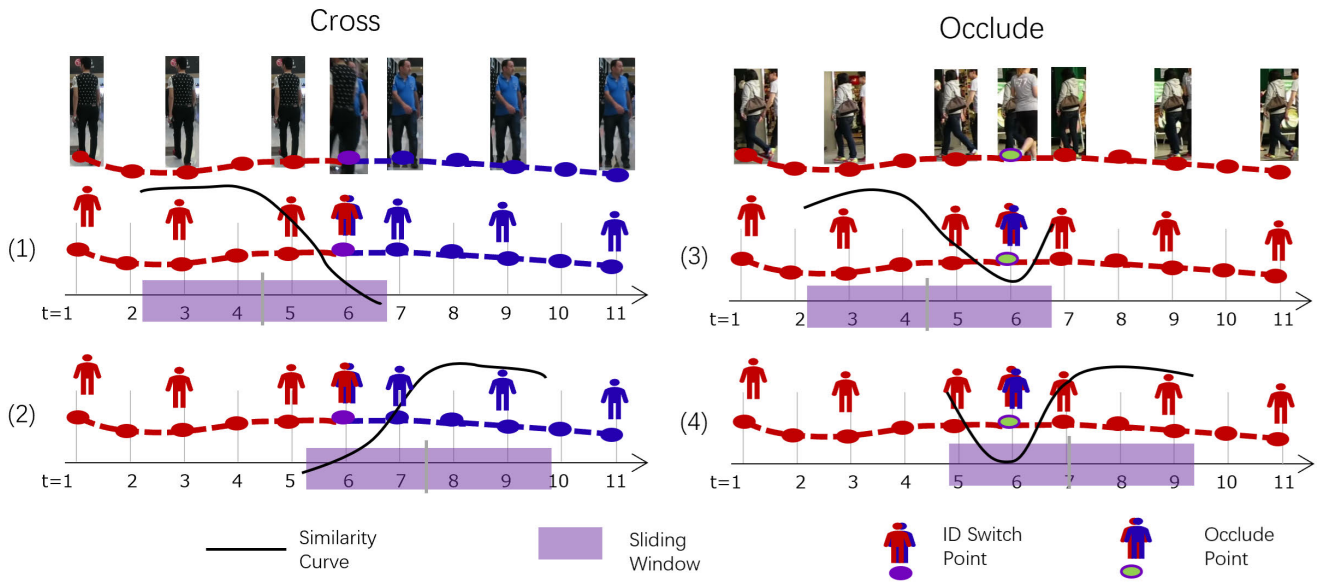

FIGURE 1. Cropped images in the first row are the selected frames in tracklet for cross/occlude scene. (1)/(2) depicts the similarity curve when the ID switch point enters and exits the sliding window. (3)/(4) depicts the similarity curve when the occlude point enters and exits the sliding window. (better view in color).

$t = 6$ in Figure 1 (1) and Figure 1 (3)) are similar for both 'cross' and 'occlude' scenes. Therefore, merely relying on the previous and current frames in online tracking refinement is often insufficient in determining whether an ID switch has occurred. Inspired by offline methods that integrate information both before and after the current moment, we propose to incorporate near-future frames within a short duration, such as 1∼2 seconds. It's important to note that this method still operates on a frame-by-frame basis and doesn't necessitate the entire trajectory, making it an online approach. This strategy strikes a balance between real-time processing and high accuracy, albeit with the trade-off of introducing some latency due to reliance on future information.

In different situations such as cross, temporary occlusion, and continuous tracking of the same person, the pedestrian appearance will exhibit specific characteristics, with which the ID switch point can be detected. Here, we adopt a short duration window to calculate the feature similarity curve within the window range by comparing the appearance features of the center frame and other frames. This approach can capture appearance changes effectively. For the 'cross' scene, the ID switch causes a change to the object's identity in the latter part of the trajectory. As the ID switch frame (e.g., $t = 6$ in Figure 1 (1)) first enters the short duration window from the right side, the appearance within the window will change drastically since it contains two different objects. Moreover, when the ID switch frame is about to exit the window from the left side as the window moves along the time axis (as shown in Figure 1 (2)), the appearance change within the window will terminate, since the window is about to be solely filled with the switched identity. Therefore, for the 'cross' scene, the appearance similarity curve within the window experiences a significant drop (Figure 1 (1)), followed by a sharp increase (Figure 1 (2)) around the ID switch frame as the window moves forward. For the 'occlude' scene, the target object is blocked by others, but it returns in the following frames. As the occluded frames enter and exit the short duration window (as shown in Figure 1 (3) and Figure 1 (4)), the appearance will have minor changes since the frames within the short duration window still contain the same object. In this case, the appearance similarity curve will have a slight drop and then increase immediately. Based on the previous analysis, the ID switch position will be detected when similarity has a sharp drop and increase.

After the ID switch position is detected, the latter frames that contains a different object will be split from the original tracklet. The split piece will then be merged with other existing tracklets. In the merging step, the similarity between the split piece and previous tracklets will be calculated to decide if the switched identity has ever occurred in the previous tracklet. Based on this analysis, the split piece will either be assigned with a new ID or merged with an existing tracklets. In our semi-online tracking refinement method, only a short duration of future frames (i.e. 1∼2 seconds) is utilized. These frames only represent one specific appearance in a given capturing condition and lack variance such as changes in lighting, pose, or background etc. When comparing the similarity with existing tracklets, which typically cover longer temporal periods, the merging process may fail if the tracklet contains appearance change. To address this issue, we use multiple prototypes to represent the tracklet, each encapsulating a typical appearance. The similarity is then computed between the short duration split piece and the tracklets' prototypes. If the split piece's

appearance feature is similar to any of the prototypes, it will be merged with the corresponding tracklet. This enhances the method's capability to handle diverse appearance changes over the tracklet's temporal span.

To the best of our knowledge, our semi-online tracking refinement method is the first of its kind, where the split and merge steps are performed in a frame-by-frame manner. This method can serve as a plug-and-play module and can be added to the existing tracker for real-time online tracklet refinement. The contributions of our method can be summarized as three aspects:

I. By integrating the near future frames in the short duration window, the 'drop-increase' characteristic of similarity curve shape around ID switch is identified to predict the split position.

II Multi-prototype representation is adopted in the merge step to ease the difficulty of comparing similarity between split piece that containing short-range appearance and the long-range tracklets, which may have appearance variances, such as changes in lighting, pose, or background, etc.

III. A semi-online tracking refinement method is proposed so that the refinement can be performed in a frame-by-frame manner. Compared with the baseline, our refinement method can significantly improve the IDF1 score on MOT17 and MOT20 in a real-time manner.

In this paper, we present a novel semi-online tracking refinement method aimed at improving MOT (multi-object tracking) accuracy. In the following sections, we first discuss the previous work in MOT and also in tracklet refinement methods that aiming to correct the ID switches, and highlight the limitations of existing methods in addressing this challenge (Section II). We then delve into the details of our proposed semi-online tracking refinement method, including tracklet split method based on similarity curve (Section III-A) and tracklet merge method based on multi-prototype representation (Section III-B). Next, we provide experimental results to evaluate the performance of our method on benchmark datasets and compare it with baseline approaches (Section IV). Additionally, we discuss the key components in our method and also the limitation and comparison with previous works (Section V). Finally, we conclude our paper by summarizing the main findings, discussing potential future research directions, and emphasizing the significance of our contributions to the field of MOT (Section VI).

II. RELATED WORKS

A. MULTI OBJECT TRACKING (MOT)

The baseline MOT trackers mainly have two categories: SORT(simple online realtime tracking) -based methods [12], [13], [14], [15], [16], [17], [18], [19], [20], [21], [22], [23], [24], [25], [26], [27], [28], [29] and transformer-based methods [30], [31], [32], [33], [34], [35]. Among the SORT-based methods, original SORT [14] uses Kalman filter to predict the object location and uses Hungarian algorithm to associate the bounding boxes. DeepSORT [12] extends the original SORT by incorporating appearance feature via a pre-trained association metric. Bytetrack [13] uses YOLOX [36] based detector and only uses the IoU for association. StrongSORT [37] improves the DeepSORT method in terms of object detection, feature embedding, and trajectory association, and can largely surpass the baseline. BoTSORT [15] combines motion and appearance and uses advanced camera-compensation and kalman filter. Recently, researches in motion compensation [24], [26], [27] and noise handling in crowd scene [22], [29] are gaining more and more attention and have largely surpass the previous online trackers. For example, UCMCTracker [18] uses projected probability distribution on ground plane to capture the motion patterns for further compensation. ConfTrack [29] adopts the combination of low score object penalization and cascading method to handle noisy detections. Both these two methods have top rank in the MOT challenges.

Recently, the attention machanism based transformer has also been widely used for MOT. TrackFormer [30] proposes an end-to-end model using the encoder-decoder transformer. Transtrack [31] proposes to do object detection and association in a single shot. MOTR [32] models the tracked instances as 'query' and uses them to perform iterative prediction. TransMOT [33] uses graph transformer to model the objects' spatial-temporal interactions.

In addition to the two major categories mentioned above, which are both online tracking methods, a smaller subset of research has utilized semi-online approaches for object detection and tracking [38], [39], [40]. These methods employ techniques such as Markov Random Field [38] or graph model [39] to achieve their objectives. Similar to our approach, they also leverage information from near-future frames for tracking purposes. However, it's important to note that our method differs from these approaches in that it serves as a post-refinement method. This means that it can be seamlessly integrated as a plug-and-play module into existing trackers, enhancing their performance without requiring extensive modification.

Furthermore, given the rapid advancements in the MOT research field, the performance of online trackers has significantly improved, achieving state-of-the-art results in the MOT17 and MOT20 challenges. Therefore, in this paper, we primarily use the online method as the baseline to demonstrate the performance improvement achieved by our semi-online refinement method.

Despite these advances on MOT tracking methods, ID switches are common, mainly due to occlusion, necessitating tracking refinement for further correction.

B. TRACKING REFINEMENT

Since ID switch is inevitable for the existing MOT tracking method, several tracking refinement methdods [1], [2], [3], [4], [5], [6], [7], [8], [9], [10], [11] have been proposed. These methods all follow the 'split-merge' procedure, where

the ID switch position is first detected and used to split the tracklet into small pieces, then the split pieces are merged into a new tracklet aiming to make sure it only contains single identity. Among these methods, [1] proposes a model-agnostic method where the split and merge steps are improved by self-supervised learning on appearance features. Reference [3] uses appearance based clustering for tracklet splitting and completion. Reference [2] uses stacked dialted convolution for the split and multi-head self-attention encoder for the merge. Recently, graph model has been widely used in both tracklet generation [41], [42], [43], [44], [45] and tracklet refinement [46]. For example, [46] uses deep network for split position prediction and uses affinity matrix to fed into a graph model for tracklet connection prediction. However, these methods are all offline methods, which means they require the entire frame sequences in the whole trajectory for the refinement purpose. To the best of our knowledge, our semi-online tracking refinement method is the first of its kind, where the split and merge steps are performed in a frame-by-frame manner. Our method can serve as a plug-and-play module and can be added to the existing tracker for realtime online tracklet refinement.

III. METHOD

We use a semi-online scheme for tracking refinement, where the future frames within a short duration (e.g., 1∼2s) is used. For current frame t, we assume the tracklets for $0 \sim t-1$ frames are all correct, since for online process, the ID will be fixed once the tracking refinement is done. We use a short duration similarity curve to characterize the pedestrian's appearance changes and identify potential ID switches. Here, the similarity curve is defined as the re-identification (ReID) feature similarities between the central frame and remaining ones in a short duration temporal window. Let t and $(t-N \sim t+N)$ be the current frame and the short window centered at t respectively, the similarity curve $\mathcal{S}^t$ is then calculated as

$$\mathcal{S}^t = \left\{ \frac{f^t \cdot f^k}{||f^t|| ||f^k||} | k = t-N, \ldots, t-1, t+1, \ldots, t+N \right\}, \tag{1}$$

where f^k denotes the ReID feature at frame k.

The similarity curve contains two halves: the former half $\mathcal{S}^{t-}$ with frame index $k < t$ and the latter half $\mathcal{S}^{t+}$ with frame index $k > t$. The pseudo code for similarity curve calculation is demonstrated in Section I in supplement.

In order to determine whether an ID switch happens at current t-th frame, the similarity curve $\mathcal{S}^t$ will be used which utilizes both previous and future frames. If an ID switch does occur, the future frames $(t+1 \sim t+N)$ will be split from the original tracklet and then be merged with other existing tracklets. The split and merge steps will be demonstrated in Section III-A and Section III-B respectively.

A. SIMILARITY CURVE BASED TRACKLET SPLIT (SCTS)

1) CHARACTERISTIC OF ID SWITCH

Figure 2 presents similarity curves for various conditions in the 'cross' scene. When there is no ID switch within a short duration window, the similarity curve illustrates the appearance changes of the same individual over a brief period. Consequently, this curve generally maintains smoothness on both sides of frame t, as depicted in Figure 2 (1). In the scenario where two people cross in the latter part of this window and an erroneous ID switch takes place, the former N frames come from the same person while the subsequent N frames involve different individuals. As a result, the similarity curve in the former part ($\mathcal{S}^{t-}$) remains smooth with higher values, but drops drastically in the latter part ($\mathcal{S}^{t+}$), reaching around 0.0 at the point when the two person is completely switched, as shown in Figure 2 (2). This sudden drop in the latter part can be represented by $\mathcal{S}^{t+}(t+1) - min(\mathcal{S}^{t+}) > th_{gap}$. Conversely, if crossing occurs in the former part of the window duration, a symmetric similarity curve is obtained, with a sudden value increase in the former part followed by smooth and higher values in the latter part, as illustrated in Figure 2 (4). This sharp increase in the former part can be represented by $\mathcal{S}^{t-}(t-1) - min(\mathcal{S}^{t-}) > th_{gap}$. When the crossing happens precisely at the center position (frame t), the former and latter frames are from different persons, while the middle frame t is a mixture of the appearance of these two individuals. In this case, the middle frame is not similar to either of the two parts, leading to a similarity curve exhibiting lower values yet maintaining a rough symmetry in both former ($\mathcal{S}^{t-}$) and latter ($\mathcal{S}^{t+}$) halves, as in Figure 2 (3).

ID switch is identified by examining the similarity curve at each time stamp along the trajectory, and it is realized by monitering three events: pre-crossing (as shown in Figure 2 (2)), crossing (as shown in Figure 2 (3)), and post-crossing (as shown in Figure 2 (4)). When the sliding window approaches the ID switch point, that is, the switch point is covered by the latter part of this window, a similarity curve similar to that in Figure 2 (2) will be first observed. This indicates that the ID switch occurs at the future time of the current time stamp, known as pre-crossing. As the window progresses further along the time axis, a similarity curve resembling Figure 2 (3) is obtained, which means that the ID switch is occurring at current time spot t, known as crossing. Finally, when the latter part of the window is switched into another person, this will lead to higher value in the latter half of similarity curve, and a similarity curve like Figure 2 (4) will be found. Till now, the ID switch process is considered finished, i.e., post-crossing, and the ID switch detection process will be terminated.

In order to detect the ID switch, we need to distinguish the 'occlude' scene (no ID switch) with 'cross' scene (an ID switch occurs). In a similar way, 'occlude' scene can also be characterized by similarity curve, where the subject is temporarily occluded and then appears again. In other word, the tracked object doesn't change within the tracklet.

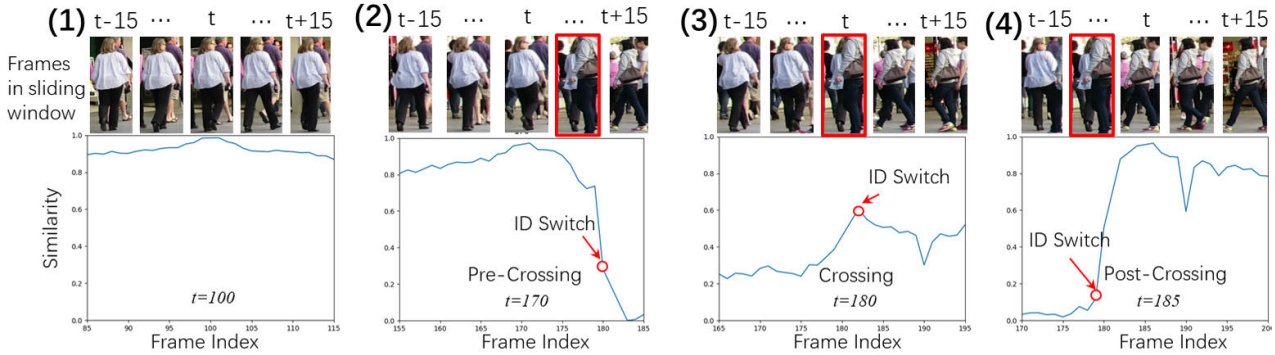

FIGURE 2. Similarity curves for 'cross' scene that is caused by ID switch. Window range is set as ($t-15 \sim t+15$). Tracklet is obtained by applying Bytetrack [13] on MOT17 video.The image with red box is the ID switch position. (1) No ID switch; (2) pre-crossing: ID switch in latter part; (3) crossing: ID switch at window center; (4) post-crossing: ID switch in former part. (better view in color).

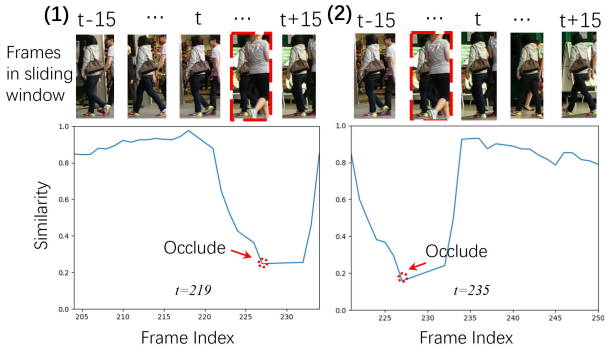

FIGURE 3. Similarity curves for 'occlude' scene that is caused by temporarily occlusion. Window range is ($t-15 \sim t+15$). Tracklet is generated by applying Bytetrack [13] on MOT17 video. Frame with red dash box is the 'occlude' position.

Therefore, the similarity curve will drop temporarily at the occluded frame since the object is blocked by others (Figure 3 (1)), and increase immediately as the object fully appears again (Figure 3 (2)). The bottom of the similarity curve is reached when the occlusion happened. At this moment, the similarity value is still relatively high since there is still some visible parts of the tracked object that have not been blocked, as shown in Figure 3 (1). In order to distinguish the similarity curve between 'cross' scene (Figure 2 (2)) and 'occlude' scene (Figure 3 (1)), the bottom value threshold th_{min} is used. If the bottom of latter half similarity curve $\mathcal{S}^{t+}$ satisfies $min(\mathcal{S}^{t+}) < th_{min}$, then it will be considered as 'cross' scene and our method will begin to search for ID switch.

2) DETECTION OF ID SWITCH POSITION

For ID switch caused by crossing, the similarity curve is characterized by several features, including sharp value drop in the former or latter part, sufficiently small minimal value, left-right symmetry, etc. In order to identify these features, we use three criteria in our method, including slope ratio, former/latter average similarity, and KL distance.

a: SLOPE RATIO

The slope ratio is defined as: $sr = sl^{t-}/sl^{t+}$. sl^{t-} is the average slope of former half of the similarity curve $\mathcal{S}^{t-}$, calculated using the curve's peak and bottom values: $sl^{t-} = |y_p^{t-} - y_b^{t-}| \,/\, |x_p^{t-} - x_b^{t-}|$. sl^{t+} can be computed in a similar manner. When transitioning from pre-crossing to crossing, the similarity curve changes from that in Figure 2 (2) to Figure 2 (3), and a large decrease of slope ratio will be observed.

Therefore, the significant drop of slope ratio may indicate an ID switch point, where :

$$sr^{t-1} - sr^{t} > th_{sr} \tag{2}$$

sr^{t-1} and sr^{t} are the slope ratio calculated using similarity value sets $\mathcal{S}^{t-1}$ and $\mathcal{S}^{t}$, where window is centered at frame $t-1$ and t respectively. Slope ratio change threshold th_{sr} is used to identify the ID switch position. The relationship between the slope ratio value and the ID switch point is illustrated in Figure 4 (1).

b: FORMER/LATTER AVERAGE SIMILARITY

Similarity value is another feature characterizing the similarity curve. When transitioning from pre-crossing (as shown in Figure 2 (2)) to crossing (as shown in Figure 2 (3)), the average value of latter part increases ($\mathcal{S}_{avg}^{t+} > \mathcal{S}_{avg}^{t-1+}$) while that of the former part decreases ($\mathcal{S}_{avg}^{t-} < \mathcal{S}_{avg}^{t-1-}$). For crossing state (as shown in Figure 2 (3)), the average values of latter and former parts are close:

$$|\mathcal{S}_{avg}^{t+} - \mathcal{S}_{avg}^{t-}| < th_{diff}, \tag{3}$$

where th_{diff} is the threshold to detect whether the average values of latter and former halves of similairty curve become close. The relationship between the former/latter average similarity value and the ID switch point is illustrated in Figure 4 (2).

c: KL DISTANCE

KL distance is adopted to measure the symmetry between former half $\mathcal{S}^{t-}$ and reversed latter half $\mathcal{S}^{t+}$ halves of similarity curve. In pre-crossing state (as shown in Figure 2 (2)), the KL distance is very large due to the drastical drop in the latter part. In crossing state (as shown in Figure 2 (3)), the similarity curve becomes symmetrical, leading to a smaller

KL distance value. With this analysis, significant drop of KL distance indicates ID switch point, where:

$$kl^{t-1} - kl^{t} > th_{kl}, \tag{4}$$

where kl^{t-1} and kl^{t} are the kl value calculated based on within-window similarity values $\mathcal{S}^{t-1}$ and $\mathcal{S}^{t}$, where window is centered at frame $t-1$ and t respectively. The relationship between the KL distance and the ID switch point is illustrated in Figure 4 (3).

These three metrics above provide a comprehensive analysis of the similarity curve, offering complementary insights: KL distance measures the symmetry of the curve, slope ratio describes its general shape, and former/latter average similarity measures its values. As a summary, the ID switch point is usually featured by the above three criteria. If the similarity curve satisfies one of the criteria above, it will be considered as ID switch point. The reason why such a loose criterion is used is that ID switch check is the foundation for the tracking refinement and it is crucial to find as many ID switch points as possible to avoid tracking performance decreasing caused by erroneous ID switch. This loose criterion may cause some over-splitting problem. However, this problem will be corrected in the following merging step since the wrongly split tracklets still have larger similarity with their original tracklet and can be merged together again. Also, the increased number of wrongly split pieces will not add much extra computational complexity for the following merge step, because the computation of similarity between split pieces and existing tracklets can run very fast, as demonstrated in Section III-B.

Besides appearance similarity, the motion information is also adopted as a complementary factor, under the assumption that pedestrian trajectorie is linear in a short time [1], [12], [14]. If the trajectory in the short duration window deviates much from a straight line, it can be thought of as the concatenation of two different trajectories. Here, the motion is formulated by a linear function $f(x) = a_0 + a_1 x$, which is approximated through least squares regression using the detection centers (x, y) of tracked object within the short duration window. For the detection center (x^t, y^t) at the middle frame t, the estimated position $f(x^t)$ is compared with the real position y^t to measure the deviation d^t from straight line. If d^t is larger than a threshold th_d, it will be considered as ID switch point:

$$d^t = ||f(x^t) - y^t||/h^t > th_d, \tag{5}$$

where h^t is the bounding box height for normalization purpose.

The pseudo code for computing metrics for split check and the workflow of the similarity curve based tracklet split are demonstrated in Section II and Section III of the supplement.

B. MULTI-PROTOTYPE TRACKLET MERGE (MPTM)

In the MOT, many tracklets are often maintained simultaneously to correspond to multiple objects. At a given moment, all these tracklets are analyzed to find ID switch points based on the similarity curve within the short duration window. If frame t is identified as ID switch point, the latter frames $(t+1 \sim t+N)$ are split from its original tracklet since the tracked object has changed. Therefore, we must decide if this divided part belong to some other tracklet or not. For this purpose, the merging operation is adopted.

For tracklet merging, the offline methods mainly use clustering [1], [11] or graph multi-cut [46] methods based on the similarity matrix among tracklet pairs. However, these methods are time-consuming and don't meet the real-time requirement of online applications. To solve this problem, we directly compare the appearance between the split part (latter frames) and existing tracklets $\mathcal{T} = \{\mathcal{T}_i | i = 1, \ldots, M\}$, and use an appearance similarity threshold to decide if the split piece should be merged with existing tracklets or be assigned with a new track ID.

Since only a short duration of future frames, i.e. 1∼2 seconds, are used in our online tracking refinement method, these frames only represent one specific appearance in a given capturing condition and lack variance such as changes in lighting, pose, or background etc. When comparing the similarity with existing tracklets which usually have long temporal coverage, the merging process may fail if the tracklet contains appearance change. To solve this problem, we use multiple prototypes to represent the tracklet, where each prototype encapsulates a typical appearance, thereby enhancing the method's capability to handle diverse appearance changes over the tracklet's temporal span.

Inspired by [47], an incremental updating scheme is designed to get multiple prototypes for each tracklet. For i-th tracklet $\mathcal{T}_i \in \mathcal{T}$, a prototype set is maintained $\mathcal{P}_i = \{p_i^j\}$. When new detection is added to $\mathcal{T}_i$, cosine similarities between its appearance feature f and the prototype set $\mathcal{P}_i$ are first calculated. If any of the similarity value is larger than a given threshold th_{sim}, then feature f will be used to update the prototype with maximum value by a moving average strategy. Otherwise, if none of the similarity values surpasses the threshold, this means that the new appearance feature is sufficiently different with respect to the previous prototypes. In this case, the current feature f will be assigned as a new prototype and added to the prototype set.

During tracklet merging, cosine similarities between the average feature of split latter frames f^+ and the prototype set $\mathcal{P}_i$ of i-th tracklet $\mathcal{T}_i$ will be computed. If the maximum value of the cosine similarities between f^+ and $\mathcal{P}_i$ is larger than a given threshold th_{sim}^{pr}, it means the object in the split latter frames has appeared before in the i-th tracklet. Then the split latter frames will be attached to i-th tracklet. Otherwise, they will be considered as never showed up in the previous tracking scene. Therefore, the split latter frames will be assigned with a new tracklet ID and the average feature f^+ will be assigned as its prototype.

In real-world scenarios, multiple tracklets may have similarity values exceeding the predefined threshold to the average feature of the split piece f^+. In such cases, the split

piece should merge with the tracklet that has the highest similarity value.

The pseudo code for multi-prototype-based tracklet merge and the overall workflow of the semi-online tracklet refinement, including split and merge steps, are presented in Sections IV and V of the supplement.

IV. EXPERIMENT RESULT

A. DATASET

Datasets of MOT17 [48] and MOT20 [49] challenges are employed for evaluation. The evaluation metrics include IDF1 score, ID Precision (IDP), ID Recall (IDR), ID Switches (IDSw), and multi-object tracking accuracy (MOTA)., as specified in [48], [49], and [50]. Among these metrics, MOTA is calculated using false positive (FP), false negative (FN) and IDSw. Since the numbers of FP and FN are much larger comparing with IDSw, therefore, MOTA is more closely related to the detection performance. IDF1 score emphasizes more on the tracklet-level association accuracy rather than the detection performance. In our method, we aim to correct ID switches and improve the temporal continuity of correctly tracked identity based on existing detections. Thus, the IDF1 score will serve as the primary evaluation criteria.

B. IMPLEMENTATION DETAILS

Since our method is based on the preacquired tracklets, which makes it eligible to be attached to any existing MOT tracking methods. Here, we select Bytetrack [13] to generate tracklets as it properly balances accuracy and speed. In order to achieve fast inference time, among the high performance ReID models [51], [52], [53], [54], [55], we choose a fast openvino model [52] to encode the appearance feature of detected object, which is based on the OmniScaleNet [51] backbone with Linear Context Transform [56] blocks developed for fast inference.

The first parameter we need to decide is the optimal value for sliding window width N. We have tested the impact of different values of N ($5 \sim 25$ with interval as 5) on MOT17 videos, as shown in Table.1. The speed in Table 1 is tested on Intel(R) Xeon(R) CPU E5-2620. In the result, the IDF1 score increases in the beginning when $N < 15$ and saturates when $N \geq 15$. The reason is that it's difficult for shorter window to capture the trend of similarity curve around ID switch. However, larger N doesn't provide additional discriminativeness for the detection and correction of ID switch and the runing speed drops significantly since more frames are utilized. For a better trade-off between the performance and speed, the window width N is selected as 15 in the following experiments, which is about 1.5 seconds given the video's frame rate. This is consistent with our intention where the future 1∼2 seconds are used to detect the ID switch.

To determine the parameters for ID switch detection (as shown in Section III-A1 and III-A2), several ID switch points on the MOT17 tracklets are manually annotated. The values

TABLE 1. Window width N selection based on MOT17. For speed (frames per second (FPS), tested on Intel(R) Xeon(R) CPU E5-2620), '$*$' means the speed for Bytetrack, including object detection and association steps. '†' means the speed for online tracking refinement method, including appearance feature extraction, ID switch detection and split-merge steps.

WindowWidth	Bytetrack	5	10	15	20	25
IDF1	79.2%	80.4%	81.1%	**81.7%**	81.7%	81.9%
Speed (FPS)	29.6^*	$83^\dagger$	$66^\dagger$	$\mathbf{54}^\dagger$	$30^\dagger$	$25^\dagger$

of the ID switch check criteria described in Section III-A2 at different annoated ID switch points, where $th_{gap} = 0.8$, $th_{min} = 0.1$, $th_d = 0.15$, $th_{kl} = 3.0$, $th_{sr} = 3.0$, and $th_{diff} = 0.25$. Since the capture environment differs from one video to another, the optimal parameter may vary. But, the ID switch point will still be found since it only has to satisfy one of the criteia. For the tracklet merging, parameters th_{sim} and th_{sim}^{pr} in Section III-B are empirically set as 0.8 and 0.6.

C. COMPARISON OF CRITERIA FOR ID SWITCH DETECTION

In order to detect the ID switch point, we propose a method to examine the similarity curve within a short-term window using three criteria, including slope ratio, former/latter average similarity and KL distance. The relationships between ID switch points and the curves of these three criteria are illustrated as Figure 4. The tracklets are generated using Bytetrack on MOT17-09 video (a sample frame for MOT17-09 is shown in Figure 5).

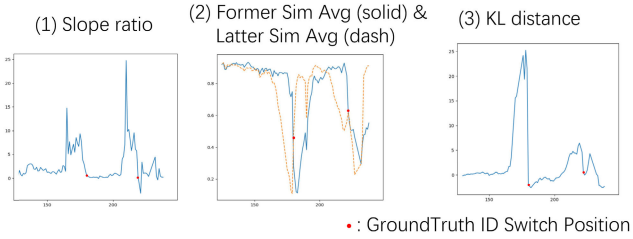

FIGURE 4. Curves of the three different criteria (slope ratio, former/latter average similarity, KL distance) for ID switch point detection. Red dot indicates the ground truth ID switch point. (better view in color).

In Figure 4, the red dots highlight the ground truth ID switch points, which are obtained by visual inspection on the original tracklet generated by Bytetrack. We can observe that the trends we have demonstrated in the Section III-A2 are consistent at the ID switch point. Specifically, at ID switch point, we can notice a sharp drop in both slope ratio (Figure 4 (1)) and KL distance (Figure 4 (3)), while the former average similarity increases, and latter average similarity decreases in Figure 4. Moreover, at the ID switch point, former/latter average similarity values become close. Therefore, we can conclude that using these three criteria to evaluate the similarity curve can facilitate the detection of the ID switch point.

Our approach adopts a loose constraint where the frame is classified as an ID switch point if any one of the three criteria is satisfied. By doing so, we aim to ensure maximum recall

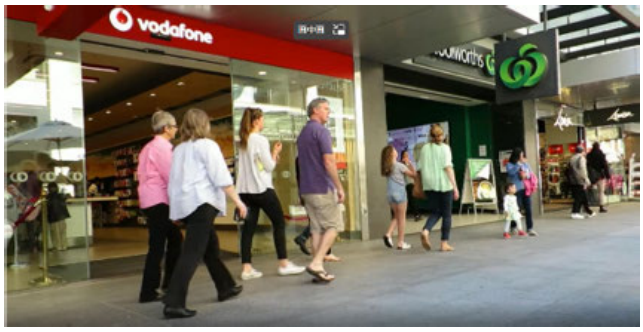

FIGURE 5. Sample frame of MOT17-09 video.

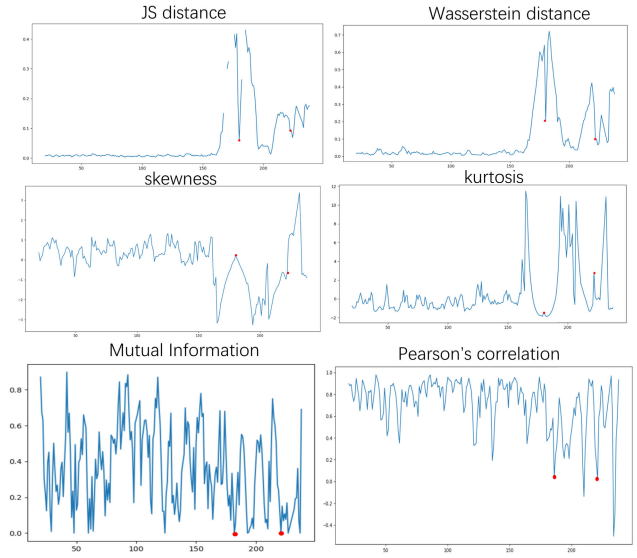

FIGURE 6. Other criteria for similarity curve based ID switch point detection. Red dot indicates the ground truth ID switch point. (better view in color).

of the real ID switches. Although this approach may lead to falsely detected ID switch points, these points will still be used for the following tracklet split and merge. Since the split pieces generated by the falsely detected ID switch points still contain same object with respect to the original tracklet, they will be reattached to the original tracklet in the merge step, since they still have larger similarity to the original tracklet.

Besides these three criteria, we have also tested other possible methods, including skewness/kurtosis of similarity curve, mutual information and correlation between former and latter halves of similarity curve, and JS distance and Wasserstein distance between former and latter similarity curve. The curves of these criteria are also generated based on the tracklets generated by Bytetrack on MOT17-09 video. However, from Figure 6, we can see that the ID switch position (red dot) doesn't show discriminative characteristic on these curves, which make it difficult for them to be utilized for ID switch detection. Therefore, in our method, we use the previously stated three criteria for the ID switch detection instead of using these methods.

TABLE 2. Ablation study for different criterion for similarity curve based tracklet split (SCTS). 'F/L sim avg' means 'former/latter average similarity'. Larger drop on IDF1 indicates larger importance of this criterion.

Dataset	Experiment	IDF1	IDP	IDR	IDSw	MOTA
MOT17	Bytetrack [13]	79.2%	83.9%	74.9%	165	**76.6%**
	SCTS	**81.9%**	**87.7%**	**76.9%**	**128**	76.4%
	- slope ratio	80.7%(-1.2)	86.1%	75.9%	131	76.3%
	- F/L sim avg	80.8%(-1.1)	86.4%	75.9%	135	76.0%
	- KL distance	80.4%(-1.5)	86.0%	75.4%	135	76.1%
	- motion	81.5%(-0.4)	87.1%	76.5%	129	76.1%
MOT20	Bytetrack [13]	75.2%	81.4%	70.2%	1,223	77.8%
	SCTS	**78.3%**	**84.1%**	**73.2%**	**1,097**	**77.9%**
	- slope ratio	77.3%(-1.0)	83.2%	71.9%	1121	77.7%
	- F/L sim avg	77.5%(-0.8)	83.0%	72.0%	1109	77.8%
	- KL distance	77.1%(-1.2)	82.7%	70.9%	1134	77.6%
	- motion	78.0%(-0.3)	83.9%	72.5%	1097	77.8%

D. ABLATION STUDY

1) ABLATION STUDY FOR ID SWITCH DETECTION CRITERIA IN SIMILARITY CURVE BASED TRACKLET SPLIT (SCTS)

Here, the criteria for ID switch detection consist of similarity curve based methods, such as slope ratio, former/ latter similarity average, and KL distance, supplemented by the motion-based metric. In order to evaluate the importance of different ID switch detection criterion in SCTS, first of all, the overall criteria combination is evaluated using MOT metrics. Then each individual criterion are elinimated from the combination to assess the performance drop, as shown in Table.2. In this way, we can find out which criterion has more contribution. The results show that the similarity curve based metrics are more important than the motion based metric since the performance drop is larger if these criteria are removed. The reason might be that the deviation of objects' moving direction is small in the MOT videos. Nonetheless, the motion based metrics may still be useful to detect the ID switch position, especially when the objects move in varying direction.

As for similarity curve-based methods, similar trends on MOT17 and MOT20 are observed, where both KL distance and slope ratio make comparable contributions towards improving the IDF1 score, since they both measure the evenness of the similarity curve. However, KL distance outperforms slope ratio slightly as it is evaluated element-wise and is more sensitive, whereas slope ratio describes the general shape of the curve. The former/latter average similarity shows a similar level of contribution as KL distance/slope ratio, indicating that the trend of similarity curve around the ID switch position cannot be fully described by the similarity curve shape alone (KL distance/slope ratio), but also by the values of the similarity curve (former/latter average similarity).

2) ABLATION STUDY FOR MULTI-PROTOTYPE REPRESENTATION IN MULTI-PROTOTYPE TRACKLET MERGE (MPTM)

In order to assess the significance of tracklet's multi-prototype representation in multi-prototype tracklet merge

TABLE 3. Ablation study for multi-prototype tracklet representation in multi-prototype tracklet merge (MPTM).

Dataset	Experiment	IDF1	IDP	IDR	IDSw	MOTA
MOT17	Bytetrack [13]	79.2%	83.9%	74.9%	165	76.6%
	TM w MP	**81.9%**	**87.7%**	**76.9%**	**128**	**76.9%**
	TM wo MP	81.3%(-0.6)	87.0%	76.3%	136	76.4%
MOT20	Bytetrack [13]	75.2%	81.4%	70.2%	1,223	77.8%
	TM w MP	**78.3%**	**84.1%**	**73.2%**	**1,097**	**77.9%**
	TM wo MP	76.3%(-2.0)	82.0%	71.5%	1,127	77.7%

(MPTM), tracklet merging with/without multi-prototype representation are evaluated, as shown in Table.3. We can observe that without multi-prototype representation, the tracklet merge performance both drop on MOT17 and MOT20. However, the drop rate in MOT20 is larger comparing with MOT17, which means multi-prototype representation plays an more important role in the MOT20 tracking refinement.

The purpose of multi-prototype strategy is to represent the typical appearance change for the long-range tracklets so that the matching with the short duration split pieces can be more likely to succeed, since the split piece only captures the appearance within only 1∼2 seconds. On MOT20, the video length is much longer comparing to MOT17 (2 minutes v.s. 30 seconds), which means it has much more long-range tracklets during the tracklet merging step. This is the reason why MPTM is more effective on MOT20.

E. REFINEMENT WITH SOTA TRACKING METHOD

Our semi-online tracklet refinement method corrects the ID switch in a frame-by-frame manner and can be added as a plug-and-play module to any existing tracking methods. In the previous experiments, Bytetrack [13] is used as the baseline tracker. Since Bytetrack doesn't use appearance feature, we also investigate several other SOTA tracking methods, including Strong-SORT [37], BoT-SORT [15], ConfTrack [29] and UCMCTrack [18] and use our semi-online method for tracklet refinement. Comparing with transformer based tracker, the selected SORT-based methods have better performance in terms of IDF1 score on MOT17/20. Therefore, we use these four models for further evaluation.

From Table.4, by using our semi-online refinement method, the tracking metrics can all be improved, especially for the ID related metrics, such as IDF1. However, the improvement on MOTA is rather small than the IDF1 score, or decrease slightly in some cases. The main reason is that MOTA is calculated based on the FP, FN and ID switch. Since ID switch number is much smaller comparing with the FP and FN, MOTA is mainly related to the detection results. In our method, the detection is same with the baseline trackers, therefore the changes of FP and FN are negligible, and the reduction of ID switch number is too small comparing to the FP and FN value. This is why the correction of ID switch contributes less to the improvement on MOTA than on IDF1 score.

TABLE 4. Refinement with state-of-the-art (SOTA) tracking methods. For the speed (frames per second (FPS), tested on Intel(R) Xeon(R) CPU E5-2620), '$*$' means the speed for the original tracking method, including object detection and association steps. '†' means the speed for the online tracklet refinement method, including appearance feature extraction, ID switch detection and tracklet split-merge. Since offline method cannot run in frame-by-frame manner, the speed (FPS) is 'n/a'.

Dataset	Experiment	IDF1	IDP	IDR	IDSw	MOTA	FPS
MOT17 Videos	Bytetrack [13]	79.2%	83.9%	74.9%	2196	76.6%	29.6*
	+ours	81.9%	87.7%	76.9%	1828	76.4%	54.0†
	+offline [1]	82.3%	88.9%	77.6%	1710	76.9%	n/a
	Strong-SORT [37]	79.5%	84.2%	74.4%	1194	79.6%	27.6*
	+ours	82.4%	87.2%	76.4%	1021	79.5%	54.0†
	+offline [1]	82.5%	89.2%	77.4%	919	80.5%	n/a
	UCMC [18]	81.0%	85.1%	76.9%	1689	80.6%	23.7*
	+ours	82.9%	88.4%	77.4%	1479	80.5%	54.0†
	+offline [1]	83.0%	88.1%	77.9%	1221	80.6%	n/a
MOT20 Videos	Bytetrack [13]	75.2%	81.4%	70.2%	1,223	77.8%	17.5*
	+ ours	78.3%	84.1%	73.2%	1,127	77.7%	33.0†
	+offline [1]	79.2%	85.0%	73.9%	1,097	77.9%	n/a
	BoT-SORT [15]	77.5%	82.5%	71.9%	1,313	77.8%	15.5*
	+ours	79.0%	85.4%	73.2%	1,197	77.8%	33.0†
	+offline [1]	79.8%	86.1%	73.9%	1,123	77.9%	n/a
	ConfTrack [29]	80.2%	83.5%	76.9%	702	77.2%	14.0*
	+ours	81.2%	85.9%	76.5%	671	77.1%	33.0†
	+offline [1]	81.5%	86.3%	76.8%	653	77.2%	n/a

To the best of our knowledge, our semi-online tracking refinement method is the first of its kind. Therefore, it is challenging to compare it with SOTA methods of the same type. Instead, we turn to an offline refinement method [1] for comparison, which employs frames from the whole trajectory. As the analysis above suggests, offline method must wait for the entire trajectory to be generated. Although it typically performs better than online methods due to the utilization of entire frame sequence, it is not suitable for online applications. Our goal in this comparison was not necessarily to surpass the performance of offline method, as our online tracking refinement method only utilizes neighboring frames within a short duration window. We sought to assess whether the drop in performance is acceptable for practical use. As seen in Table.4, our online method ('ours') slightly underperforms the offline method [1], but achieves a substantially fast processing speed. It should be noted that the performance drop is relatively minor, with an average decrease of only 0.5% on IDF1 score, which is acceptable for practical application.

F. EXAMPLE OF ID SWITCH CORRECTION USING OUR SEMI-TRACKING REFINEMENT METHOD

One example of the ID switch correction is shown in Figure 7. Here, the first row is the pre-obtained tracklet where two different objects are assigned with same tracklet ID (ID 4). After tracklet refinement using our semi-online method, the new object will be assigned as a new tracklet (ID 26) since it has never occured in the previous frames. Since the original tracking result contains multiple track IDs, for better visualization, we only select one target object here for demonstration.

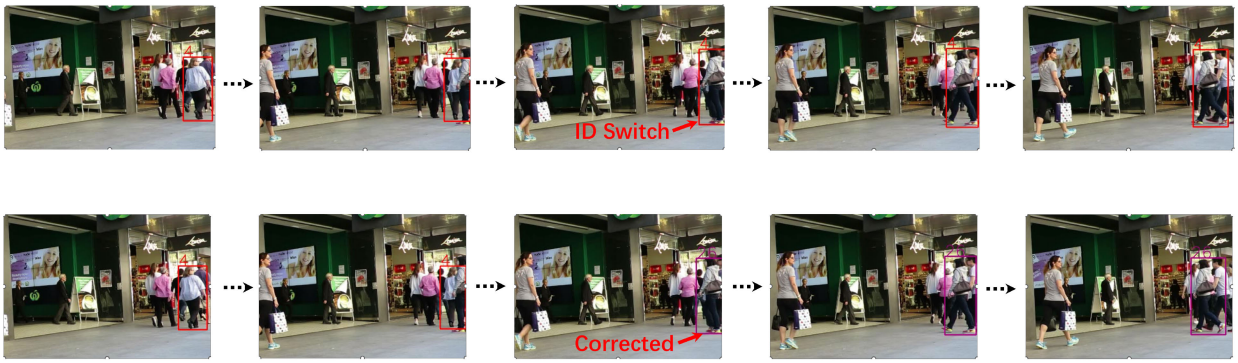

FIGURE 7. Example of correction on MOT17 video using our tracklet refinement method. First row is the pre-obtained tracklet by Bytetrack that contains ID switch. Second row is the tracklet refined by our online method. The frames that contain ID switch and after correction have been annotated with red arrow. (better view in color).

V. DISCUSSION

The key advantage of the proposed semi-online tracking refinement method over traditional approaches lies in its ability to process temporal data in real-time while still achieving high accuracy. Unlike traditional offline methods that require the entire trajectory for refinement, the semi-online method incorporates only a short duration of future frames. This enables it to make decisions frame-by-frame, allowing for faster processing and more immediate feedback. Additionally, by integrating near-future frames within a short duration window, the method can accurately predict ID switch points without relying on the entire trajectory, striking a balance between real-time processing and high accuracy.

The method detects the ID switch point by monitoring changes in appearance similarity within a short temporal window. When an ID switch occurs, frames containing different objects enter the window, causing a significant drop in appearance similarity followed by a sharp increase as the window moves forward. The ''drop-increase'' pattern in appearance similarity within the moving temporal window is identified through the analysis of several metrics, including slope ratio, KL divergence, average similarity values, and motion deviation within sliding windows. This analysis aids in pinpointing the ID switch point.

This step is crucial for effective tracklet split and subsequent merging because it accurately identifies the point where a change in object identity occurs within a tracklet. By precisely segmenting the tracklet at the ID switch point, the method ensures that frames containing different objects are split from the original tracklet. This segmentation is essential for maintaining the integrity of each tracklet and avoiding ambiguity in subsequent processing steps.

Furthermore, by detecting the ID switch point, the method can effectively merge split tracklets based on similarity criteria, ensuring that frames containing the same object are grouped together. This merging process helps reconstruct accurate trajectories by connecting split tracklets and minimizing fragmentation in the tracking results. Overall, the detection and utilization of the ID switch point are fundamental steps that enable the method to achieve accurate tracklet segmentation and merging, leading to improved multi-object tracking performance.

While a comprehensive review of related work is provided in the 'Related Work' section of our paper (Section II), it's essential to briefly discuss previous approaches in this context. Prior refinement methods in multi-object tracking have predominantly focused on offline tracklet refinement techniques, where the entire trajectory is processed retrospectively to detect and correct ID switches. While these methods have demonstrated efficacy in improving tracking accuracy, they often suffer from limited scalability and real-time processing constraints. In contrast, our proposed semi-online tracking refinement method offers a novel approach by combining elements of both online and offline processing, allowing for more efficient and adaptive tracklet refinement in real-time tracking scenarios. By leveraging near future frames within a short temporal window, our method effectively identifies and addresses ID switches without the computational overhead associated with processing the entire trajectory. This hybrid approach not only enhances tracking accuracy but also improves computational efficiency, making it well-suited for real-world multi-object tracking applications.

While our proposed semi-online tracking refinement method shows promising results in improving multi-object tracking accuracy, it is not without its limitations. One notable limitation is the reliance on appearance similarity for ID switch detection, which may not always accurately capture the underlying object identity changes, particularly in scenarios with complex occlusions or abrupt appearance variations. Additionally, our method may struggle in handling long-term occlusions or object disappearances, as it primarily focuses on fixed short-term temporal windows for tracklet refinement. Addressing these limitations will be crucial for further enhancing the robustness and applicability of our approach in real-world multi-object tracking scenarios.

VI. CONCLUSION

In this work, we proposed an semi-online tracking refinement by using the near future frames with in 1∼2 seconds.

It does not require the whole trajectory but just the local neighbouring frames within a short duration window. This makes our method capable of being conducted in a frame-by-frame manner. Our method follows the two-step refinement procedure including split and merge. For split step, it will segment the tracklet into pieces at the frame of ID switch and ensure each piece is only related to one person. To detect the ID switch point, a sliding window is running across the tracklet to capture the similarity change around ID switch position. Around the ID switch frame, the latter half of the similarity curve will first drop sharply since the latter half frames of the window contain different object as the ID switch frame enters the window. As the short duration window moves forward, the former half of the similarity curve will increase when the ID switch frame is about to exit the window. When the window centers at the ID switch frame, the similarity curve will become even. Based on this pattern, the ID switch point can be detected and utilized for the tracklet splitting. Once the tracklet is segmented, the short duration frames are used to compare appearance similarity with existing tracklet for merging. Our method utilizes multiple prototypes to represent the tracklet since the short duration latter frames only capture a very small period of appearance and may miss important long-term appearance variances, such as changes in lighting, pose, or background etc. Our method shows promising improvement both on MOT17 and MOT20.

REFERENCES

[1] F. Yang, X. Chang, S. Sakti, Y. Wu, and S. Nakamura, "ReMOT: A model-agnostic refinement for multiple object tracking," *Image Vis. Comput.*, vol. 106, Feb. 2021, Art. no. 104091.

[2] G. Wang, Y. Wang, R. Gu, W. Hu, and J.-N. Hwang, "Split and connect: A universal tracklet booster for multi-object tracking," *IEEE Trans. Multimedia*, vol. 25, pp. 1256–1268, 2023.

[3] A. Specker, L. Florin, M. Cormier, and J. Beyerer, "Improving multi-target multi-camera tracking by track refinement and completion," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. Workshops (CVPRW)*, Jun. 2022, pp. 3199–3209.

[4] K.-S. Yang, Y.-K. Chen, T.-S. Chen, C.-T. Liu, and S.-Y. Chien, "Tracklet-refined multi-camera tracking based on balanced cross-domain re-identification for vehicles," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. Workshops (CVPRW)*, Jun. 2021, pp. 3978–3987.

[5] F. Yang, Z. Wang, Y. Wu, S. Sakti, and S. Nakamura, "Tackling multiple object tracking with complicated motions—Re-designing the integration of motion and appearance," *Image Vis. Comput.*, vol. 124, Aug. 2022, Art. no. 104514.

[6] G. Wang, Y. Wang, H. Zhang, R. Gu, and J.-N. Hwang, "Exploit the connectivity: Multi-object tracking with TrackletNet," in *Proc. 27th ACM Int. Conf. Multimedia*, Oct. 2019, pp. 482–490.

[7] Y. Wu, H. Sheng, S. Wang, Y. Liu, Z. Xiong, and W. Ke, "Group guided data association for multiple object tracking," in *Proc. 16th Asian Conf. Comput. Vis.* Cham, Switzerland: Springer, 2023, pp. 485–500.

[8] Y. Liu, X. Zhang, B. Zhang, X. Zhang, S. Wang, and J. Xu, "Multi-camera vehicle tracking based on occlusion-aware and inter-vehicle information," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. Workshops (CVPRW)*, Jun. 2022, pp. 3256–3263.

[9] A. Specker, D. Stadler, L. Florin, and J. Beyerer, "An occlusion-aware multi-target multi-camera tracking system," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. Workshops (CVPRW)*, Jun. 2021, pp. 4168–4177.

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

[11] A. Specker, L. Florin, M. Cormier, and J. Beyerer, "Improving multi-target multi-camera tracking by track refinement and completion," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. Workshops (CVPRW)*, Jun. 2022, pp. 3198–3208.

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

[13] Y. Zhang, P. Sun, Y. Jiang, D. Yu, F. Weng, Z. Yuan, P. Luo, W. Liu, and X. Wang, "ByteTrack: Multi-object tracking by associating every detection box," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, Sep. 2022, pp. 1–21.

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

[15] N. Aharon, R. Orfaig, and B.-Z. Bobrovsky, "BoT-SORT: Robust associations multi-pedestrian tracking," 2022, *arXiv:2206.14651*.

[16] J. Cao, J. Pang, X. Weng, R. Khirodkar, and K. Kitani, "Observation-centric SORT: Rethinking SORT for robust multi-object tracking," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2023, pp. 9686–9696.

[17] G. Maggiolino, A. Ahmad, J. Cao, and K. Kitani, "Deep OC-SORT: Multi-pedestrian tracking by adaptive re-identification," 2023, *arXiv:2302.11813*.

[18] K. Yi, K. Luo, X. Luo, J. Huang, H. Wu, R. Hu, and W. Hao, "UCMCTrack: Multi-object tracking with uniform camera motion compensation," 2023, *arXiv:2312.08952*.

[19] M. Babaee, Z. Li, and G. Rigoll, "A dual CNN–RNN for multiple people tracking," *Neurocomputing*, vol. 368, pp. 69–83, Nov. 2019.

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

[21] P. Dendorfer, V. Yugay, A. Osep, and L. Leal-Taixé, "Quo Vadis: Is trajectory forecasting the key towards long-term multi-object tracking?" in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 35, 2022, pp. 15657–15671.

[22] Y. Du, J. Wan, Y. Zhao, B. Zhang, Z. Tong, and J. Dong, "GIAOTracker: A comprehensive framework for MCMOT with global information and optimizing strategies in VisDrone 2021," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. Workshops (ICCVW)*, Oct. 2021, pp. 2809–2819.

[23] C. Feichtenhofer, A. Pinz, and A. Zisserman, "Detect to track and track to detect," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Oct. 2017, pp. 3057–3065.

[24] S. Han, P. Huang, H. Wang, E. Yu, D. Liu, and X. Pan, "MAT: Motion-aware multi-object tracking," *Neurocomputing*, vol. 476, pp. 75–86, Mar. 2022.

[25] J. Seidenschwarz, G. Brasó, V. C. Serrano, I. Elezi, and L. Leal-Taixé, "Simple cues lead to a strong multi-object tracker," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2023, pp. 13813–13823.

[26] C. Xiao, Q. Cao, Y. Zhong, L. Lan, X. Zhang, Z. Luo, and D. Tao, "MotionTrack: Learning motion predictor for multiple object tracking," 2023, *arXiv:2306.02585*.

[27] F. Yang, S. Odashima, S. Masui, and S. Jiang, "Hard to track objects with irregular motions and similar appearances? Make it easier by buffering the matching space," in *Proc. IEEE/CVF Winter Conf. Appl. Comput. Vis. (WACV)*, Jan. 2023, pp. 4788–4797.

[28] X. Zhou, V. Koltun, and P. Krähenbühl, "Tracking objects as points," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, 2020, pp. 474–490.

[29] H. Jung, S. Kang, T. Kim, and H. Kim, "ConfTrack: Kalman filter-based multi-person tracking by utilizing confidence score of detection box," in *Proc. IEEE/CVF Winter Conf. Appl. Comput. Vis. (WACV)*, Jan. 2024, pp. 6583–6592.

[30] T. Meinhardt, A. Kirillov, L. Leal-Taixé, and C. Feichtenhofer, "TrackFormer: Multi-object tracking with transformers," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2022, pp. 8834–8844.

[31] P. Sun, J. Cao, Y. Jiang, R. Zhang, E. Xie, Z. Yuan, C. Wang, and P. Luo, "TransTrack: Multiple object tracking with transformer," 2020, *arXiv:2012.15460*.

[32] F. Zeng, B. Dong, Y. Zhang, T. Wang, X. Zhang, and Y. Wei, "MOTR: End-to-end multiple-object tracking with transformer," in *Proc. 17th Eur. Conf. Comput. Vis. (ECCV)*, Tel Aviv, Israel. Cham, Switzerland: Springer, Oct. 2022, pp. 659–675.

[33] P. Chu, J. Wang, Q. You, H. Ling, and Z. Liu, "TransMOT: Spatial–temporal graph transformer for multiple object tracking," in *Proc. IEEE/CVF Winter Conf. Appl. Comput. Vis. (WACV)*, Jan. 2023, pp. 4859–4869.
[34] J. Yang, H. Ge, S. Su, and G. Liu, "Transformer-based two-source motion model for multi-object tracking," *Appl. Intell.*, vol. 52, no. 9, pp. 9967–9979, Jul. 2022.
[35] Y. Zhang, T. Wang, and X. Zhang, "MOTRv2: Bootstrapping end-to-end multi-object tracking by pretrained object detectors," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2023, pp. 22056–22065.
[36] Z. Ge, S. Liu, F. Wang, Z. Li, and J. Sun, "YOLOX: Exceeding YOLO series in 2021," 2021, *arXiv:2107.08430*.
[37] Y. Du, Z. Zhao, Y. Song, Y. Zhao, F. Su, T. Gong, and H. Meng, "StrongSORT: Make DeepSORT great again," *IEEE Trans. Multimedia*, vol. 25, pp. 8725–8737, Jan. 2023.
[38] L. Lan, X. Wang, G. Hua, T. S. Huang, and D. Tao, "Semi-online multi-people tracking by re-identification," *Int. J. Comput. Vis.*, vol. 128, no. 7, pp. 1937–1955, Jul. 2020.
[39] J. Wang, Y. Guo, X. Tang, Q. Hu, and W. An, "Semi-online multiple object tracking using graphical tracklet association," *IEEE Signal Process. Lett.*, vol. 25, no. 11, pp. 1725–1729, Nov. 2018.
[40] N. Q. Ly, T. T. Nguyen, T. C. Vong, and C. V. Than, "The new high-performance face tracking system based on detection-tracking and tracklet-tracklet association in semi-online mode," *Int. J. Adv. Comput. Sci. Appl.*, vol. 11, no. 3, pp. 1–12, 2020.
[41] O. Cetintas, G. Brasó, and L. Leal-Taixé, "Unifying short and long-term tracking with graph hierarchies," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2023, pp. 22877–22887.
[42] N. Marinello, M. Proesmans, and L. Van Gool, "TripletTrack: 3D object tracking using triplet embeddings and LSTM," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. Workshops (CVPRW)*, Jun. 2022, pp. 4499–4509.
[43] Q. Liu, Q. Chu, B. Liu, and N. Yu, "GSM: Graph similarity model for multi-object tracking," in *Proc. 29th Int. Joint Conf. Artif. Intell.*, Jul. 2020, pp. 530–536.
[44] A. Girbau, F. Marqués, and S. Satoh, "Multiple object tracking from appearance by hierarchically clustering tracklets," 2022, *arXiv:2210.03355*.
[45] A. Rangesh, P. Maheshwari, M. Gebre, S. Mhatre, V. Ramezani, and M. M. Trivedi, "TrackMPNN: A message passing graph neural architecture for multi-object tracking," 2021, *arXiv:2101.04206*.
[46] D. M. H. Nguyen, R. Henschel, B. Rosenhahn, D. Sonntag, and P. Swoboda, "LMGP: Lifted multicut meets geometry projections for multi-camera multi-object tracking," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2022, pp. 8856–8865.
[47] Z. Zhong, H. Sheng, Y. Zhang, Y. Wu, J. Chen, and W. Ke, "Spatio-temporal correlation graph for association enhancement in multi-object tracking," in *Proc. Int. Conf. Knowl. Sci., Eng. Manage.* Cham, Switzerland: Springer, 2019, pp. 394–405.
[48] A. Milan, L. Leal-Taixe, I. Reid, S. Roth, and K. Schindler, "MOT16: A benchmark for multi-object tracking," 2016, *arXiv:1603.00831*.
[49] P. Dendorfer, H. Rezatofighi, A. Milan, J. Shi, D. Cremers, I. Reid, S. Roth, K. Schindler, and L. Leal-Taixé, "MOT20: A benchmark for multi object tracking in crowded scenes," 2020, *arXiv:2003.09003*.
[50] L. Leal-Taixé, A. Milan, I. Reid, S. Roth, and K. Schindler, "MOTChallenge 2015: Towards a benchmark for multi-target tracking," 2015, *arXiv:1504.01942*.
[51] K. Zhou, Y. Yang, A. Cavallaro, and T. Xiang, "Omni-scale feature learning for person re-identification," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2019, pp. 3701–3711.
[52] *OpenVINO ReID retail0277*. Accessed: Aug. 1, 2023. [Online]. Available: https://docs.openvino.ai/latest/omz_models_model_person_reidentification_retail_0277.html
[53] W. Chen, X. Xu, J. Jia, H. Luo, Y. Wang, F. Wang, R. Jin, and X. Sun, "Beyond appearance: A semantic controllable self-supervised learning framework for human-centric visual tasks," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2023, pp. 15050–15061.
[54] V. Somers, C. D. Vleeschouwer, and A. Alahi, "Body part-based representation learning for occluded person re-identification," in *Proc. IEEE/CVF Winter Conf. Appl. Comput. Vis. (WACV)*, Jan. 2023, pp. 1613–1623.
[55] S. Li, L. Sun, and Q. Li, "CLIP-ReID: Exploiting vision-language model for image re-identification without concrete text labels," in *Proc. AAAI Conf. Artif. Intell.*, vol. 37, 2023, pp. 1405–1413.
[56] D. Ruan, J. Wen, N. Zheng, and M. Zheng, "Linear context transform block," in *Proc. AAAI Conf. Artif. Intell.*, vol. 34, 2020, pp. 5553–5560.

MENGJIAO WANG received the Ph.D. degree in biomedical engineering from Tsinghua University, Beijing, China, in 2014. Since 2016, he has been a Researcher with Fujitsu Research and Development Center Company Ltd., Beijing. His research interest includes face detection and recognition.

RUJIE LIU received the B.S., M.S., and Ph.D. degrees in electronic engineering from Beijing Jiaotong University, in 1995, 1998, and 2001, respectively. Since then, he has been a Researcher with Fujitsu Research and Development Center Company Ltd., Beijing, China. He has published more than 40 articles and tens of inventions. His research interests include AI, pattern recognition, and image processing.

SEPTIANA LINA received the Bachelor of Engineering (B.Eng.) degree in electronic engineering from Satya Wacana Christian University, Indonesia, in 2007, the Master of Science (M.Sc.) degree in electrical engineering and computer science from Chung Yuan Christian University, Taiwan, in 2013, and the Doctor of Engineering (D.Eng.) degree in information and communication engineering from Tokyo Institute of Technology, Japan, in 2020. From 2007 to 2020, she was a Researcher in the university and an information-communication industry, with a research interest specifically in image analysis, computer vision, and artificial intelligence. Since 2020, she has been a Researcher with Fujitsu Laboratories Ltd. Her current research interests include the biometric field and its related technology.

NARISHIGE ABE received the B.S. degree in engineering from Osaka City University, in 2005, and the M.S. degree in information science from Osaka University, in 2007. Since 2007, he has been with Fujitsu Laboratories Ltd. He was a Visiting Scholar with Stanford University, from 2013 to 2014. His research interests include image processing, machine learning, and biometric authentication algorithms. He received the OHM Technology Award, in 2017.

SHIGEFUMI YAMADA received the B.S. and M.S. degrees in engineering from Keio University, in 1998 and 2000, respectively. Since 2000, he has been with Fujitsu Laboratories Ltd. He was a Visiting Scholar with West Virginia University, in 2006. His research interests include image processing, stochastics, and biometrics. He received the OHM Technology Award, in 2017.