Multi-Camera Tracking By Candidate Intersection Ratio Tracklet Matching

Yun-Lun Li [1] Zhi-Yi Chin [2] Ming-Ching Chang [3] Chen-Kuo Chiang [1]

[1] National Chung Cheng University, Taiwan
[2] National Yang Ming Chiao Tung University, Taiwan
[3] University at Albany – State University of New York, NY, USA

Abstract

Multi-camera vehicle tracking at the city scale is an essential task in traffic management for smart cities. Large-scale video analytics is challenging due to the vehicle variabilities, view variations, frequent occlusions, degraded pixel quality, and appearance differences. In this work, we develop a multi-target multi-camera (MTMC) vehicle tracking system based on a newly proposed Candidates Intersection Ratio (CIR) metric that can effectively evaluate vehicle tracklets for matching across views. Our system consists of four modules: (1) Faster-RCNN vehicle detection, (2) detection association based on re-identification feature matching, (3) single-camera tracking (SCT) to produce initial tracklets, (4) multi-camera vehicle tracklet matching and re-identification that creates longer, consistent tracklets across the city scale. Based on popular DNN object detection and SCT modules, we focus on the development of tracklet creation, association, and linking in SCT and MTMC. Specifically, SCT filters are proposed to effectively eliminate unreliable tracklets. The CIR metric improves robust vehicle tracklet linking across visually distinct views. Our system obtains IDF1 score of 0.1343 on the AI City 2021 Challenge Track 3 public leaderboard.

1. Introduction

With recent advancement of AI computer vision, large-scale video analytics for traffic management at a city level is now possible. Real-time multi-target, multi-camera (MTMC) vehicle tracking can provide rich information for automatic traffic monitoring and management. Through the pervasively deployed traffic cameras, automatic video analytics can improve traffic infrastructure design and congestion handling [15].

Most approaches in MTMC adopt a *tracking by detection* scenario for multi-camera vehicle detection and tracking, as in Fig. 1. After vehicle detection via widely-used deep neural network (DNN) detectors, the next step is on how best to form vehicle tracklets in each view, as well as how best to associate tracklets of the same vehicles across views. Large-scale automatic video analytics systems must handle large variability of vehicle types and appearances to meet the accuracy and reliability requirements in the real world. For applications such as vehicle re-identification, large view variations cast a significant challenge in vehicle re-identification across views. Similarly, how best to perform space-time vehicle tracklet association across views is important for vehicle counting and traffic analysis.

In MTMC vehicle tracking, the aforementioned factors of detection and tracking need to be considered jointly with additional factors such as camera synchronization, re-identification, traffic regulations, computational efficiency, power consumption and the availability of annotated data that further increase complication. Also note that how best to perform MTMC tracklet linking is essentially the re-identification of vehicles across views.

Existing works [17, 9] only rely on simple Euclidean distance or cosine similarity to calculate tracklet association across cameras. Also, only the best matching score was considered in their strategy for pairwise tracklet association. It may lead some grouping errors if the Re-ID feature is not robust enough. To address this problem, we propose a new **Candidate Intersection Ratio (CIR)** metric which can better leverage the ranking information in evaluating the connectivity between tracklets across cameras.

In this paper, we perform MTMC vehicle tracking via a bottom-up *tracklet matching* strategy. Based on state-of-the-art DNN vehicle detections[19], single-camera tracking (SCT) is first performed in each camera view to create initial tracklets. We propose three tracklet filtering strategies to fine-select reliable tracklets for consideration. Namely, we propose the **CIR tracklet matching algorithm** to associate tracklets across views with better feature aggregation. The CIR metric evaluate the similarity between tracklets with entire ranking information. The matching algorithm clusters these tracklets as a tree structure. Our method can handle large appearance variabilities of the vehicles from vari-

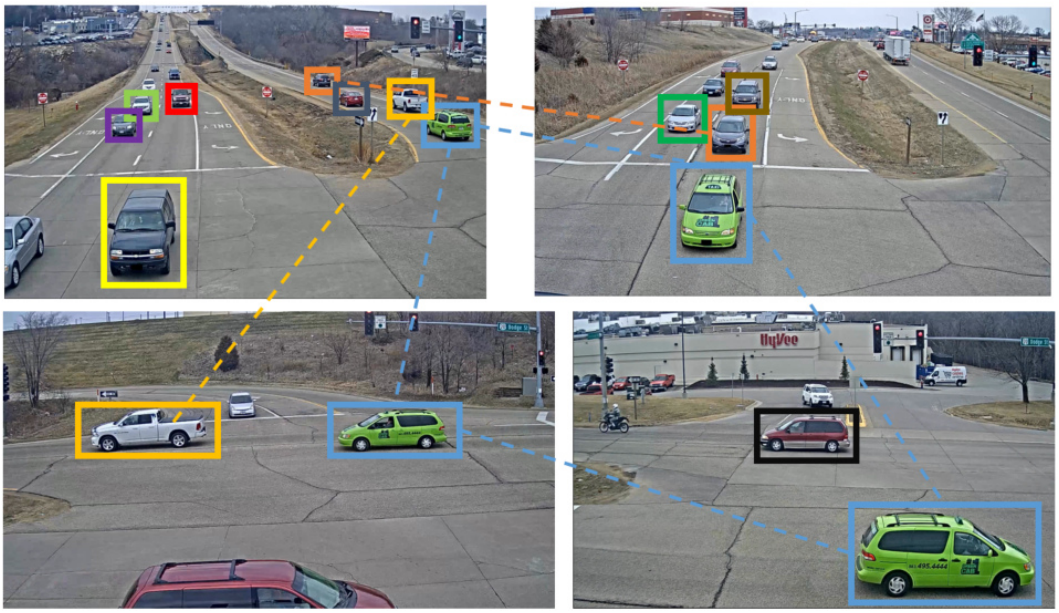


Figure 1. We target at the **Track 3 Challenge on the multi-target multi-camera (MTMC) Vehicle Tracking of the AI City Challenge 2021**. Based on the provided vehicle detection and single-camera tracking (SCT) baselines, the task in this challenge is on how best to association vehicle tracklets across a large number of camera views, while minimizing the erroneous matching of vehicles of very similar appearances as well as keeping correct matches of vehicles from very distinctive views. We propose the *Candidate Intersection Ratio (CIR) metric* and a *CIR tracklet matching algorithm* to effectively referencing to the feature ranking for better tracklet association.

ous viewing angles, distances, and brightness. Our pipeline is efficient, as the overall MTMC optimization can be decomposed into the individual SCT problems and the tracklet association problem across views.

2. Related Work

Vehicle detection. Visual object detection is an extremely active field in computer vision since the blooming of deep learning. See survey in [10]. The extensive amount of literature can be organize into two categories based on their network architecture: *two-stage* proposal driven and *one-stage* (single-shot) approaches. In general, two-stage methods (*e.g.*, Faster-RCNN [19]) can achieve high detection accuracy, and one-stage methods (*e.g.*, SSD [13], YOLOv4 [1]) can run faster.

Single-camera tracking (SCT). Many MTMC tracking methods are based on the *tracking-by-detection* [4] schemes, *e.g.*, [2, 3, 8]. Methods based on graph models [20] solve detection association problem by minimizing a designed total cost induced by the graph. Methods in this category differ in how the graph is represented. In [20], detections are treated as graph vertices, while in [22], tracklets are treated as graph vertices. In general, *detection-graph* based approaches (methods that treat each detections as a graph vertex) may encounter two problems. (1) A fundamental assumptions is the independence of each graph vertex (*i.e.* the detection across space and time). However, we know the detected object should not be conditionally independent across frames. Vehicles are frequently moving with constant speed. Therefore, temporal continuity supported by physical models should be leveraged effectively. (2) The affinity matrix representation of a detection-tracking graph is usually with high dimensionality, which makes it challenging to find the globally optimal solution. In comparison, *tracklet-graph* based approaches (methods that treat each tracklet as a graph vertex) can exploit the trajectory information to better estimate the tracklet association relationship. With proper handling of tracklet generation, even short tracklets can greatly improve the tracking association robustness and computational speed.

Vehicle re-identification (Re-ID). Object re-identification is the task of matching and searching for targets in different scenes. Re-ID features that are robust against occlusion and viewpoint changes can also play an important role in tracklet formation and matching in MTMC. The literature contains many works on person Re-ID. In [7], a strong bag-of-tricks (BoT-BS) person Re-ID baseline are presented, where features are first extracted, and preliminary ranking are obtained after feature-based sorting. Afterwards, weighted tracklet-feature based re-ranking is used to produce the final Re-ID results. In [5], an efficient, end-to-end, fully convolutional Siamese network computes the similarities at multiple levels to train a person Re-ID model. Vehicle Re-ID has wide applications in smart transportation. In [12], information other than image

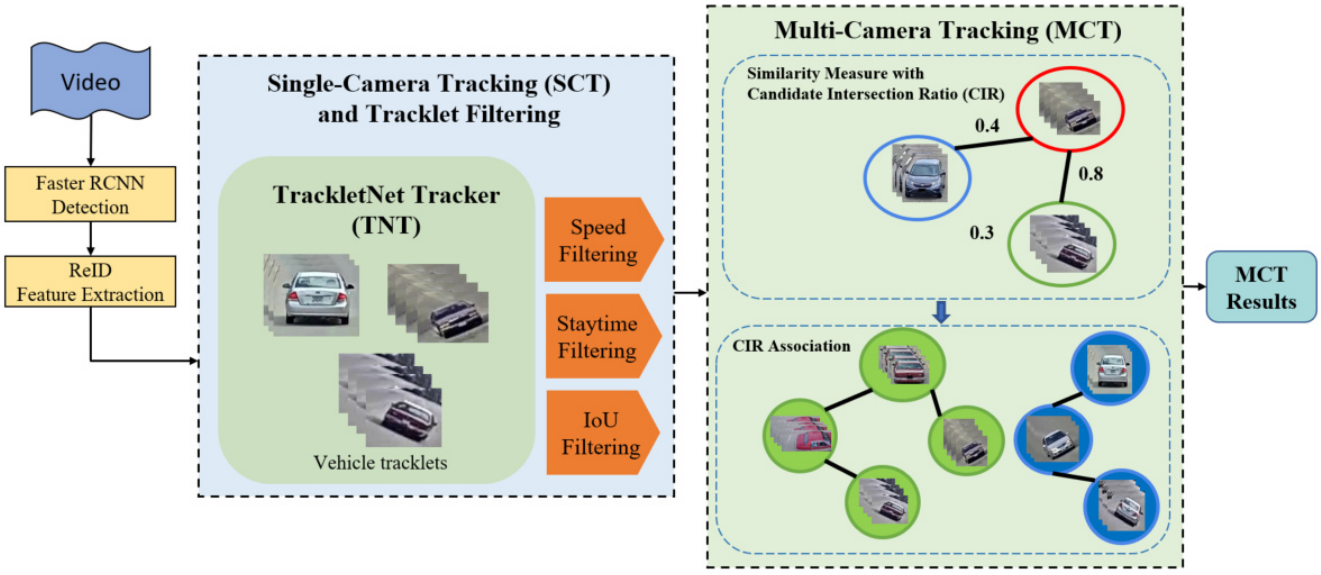


Figure 2. **The proposed multi-camera vehicle tracking pipeline.** Three tracker filters are added to effectively rule out unsuitable vehicle tracklets generated by the TrackletNet tracker. In MCT, CIR similarity are calculated to measure the suitability to link tracklet across views, which are used by a tree-based association algorithm for tracklet association. The result is a set of clustered forest of tracklets, where each tree root represent an unique tracklet ID.

appearance features (such as vehicle license plates and car model) are incorporated to improve vehicle Re-ID accuracy.

Multi-camera tracking (MCT) methods perform tracking and tracklet association jointly across multiple cameras. The multi-camera visual tracking pipeline in [14] estimates similarity terms considering appearance and dynamic motion. The General Multi-view Tracking (GMT) framework in [24] estimates the loss of the tracking targets by predicting cross-camera trajectories. To reduce the large search and matching space in MCT, [11, 21, 22] also considered camera link models with spatio-temporal constraints. For example, the unsupervised estimation process in [11] use the two-way transition time distribution. In [21, 22], vehicle speed estimation is used to establish the transition time distribution for each connected pair of vehicle across cameras. Such reliable camera link model can significantly improve cross-camera association accuracy, since the search space for matching is effectively reduced.

3. Method

The proposed multi-camera vehicle detection and tracking pipeline consists of five main modules as shown in Fig. 2: (1) vehicle detection using Faster R-CNN [19] (§ 3.1), (2) vehicle Re-ID feature extraction using `ResNet101-ibn-a` (§ 3.4), (3) Single-Camera Tracking (SCT) using the TrackletNet Tracker (TNT) tracker [23] for initial vehicle tracklet creation. (§ 3.2), (4) SCT tracklet filtering to remove unreliable tracklet predictions (§ 3.3), and (5) multi-camera tracking using the proposed Candidate Intersection Ratio (CIR) tracklet association (§ 3.5). The proposed CIR metric can effectively evaluate the similarity of each tracklets for improved association of tracklets across views. We next describe detailed steps.

3.1. Vehicle Detection

Vehicle detection baselines from mainstream DNN models including Mask-RCNN [6], SSD512 [13], YOLOv3 [18] are provided by the AI City Challenge organization. The Mask-RCNN consists of Faster R-CNN [19] and the mask module. Here we only uses the Faster R-CNN results as vehicle detections.

Faster R-CNN is still considered one of the best object detectors. On top of the design of convolution feature maps, conv layers are added to build a regional proposal network (RPN), which simultaneously outputs region bounds and objectness score for each location. Thus, RPN is a full convolution network (FCN) that can be trained end-to-end to generate high-quality region proposals, which are then sent to the Fast R-CNN for detection. The input to RPN is the raw image, and the output is a set of rectangular proposals (vehicles), each with a target score. This method slides a small network on the last shared convolution feature map to generate region proposals. This network is fully connected to an 3×3 spatial window of the input feature map. It maps Each sliding window is mapped to a low-dimensional vector, which is the input to two fully connected layers, a regression layer, and a classification layer.

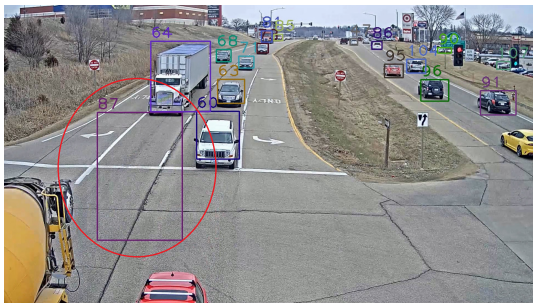
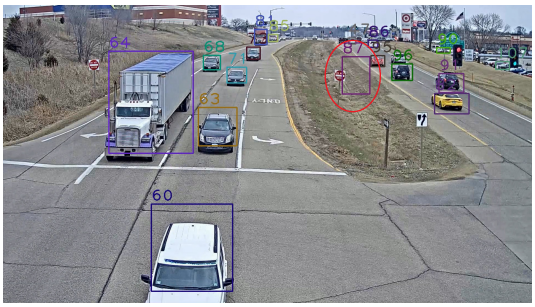
(a)  (b) 

Figure 3. Visual illustration of commonly observed problems produced by the TNT tracker [23]. (a) Detection box without an actual vehicle. (b) an empty and fast "floating" detection box that typically move at an abnormally high speed.

3.2. Single-Camera Tracking (SCT)

We adopt the TrackletNet Tracker (TNT) [23], a graph-based tracklet model for SCT. TNT takes per-frame vehicle detections as input and performs SCT based on the following three components: (1) trajectory generation, (2) trajectory connectivity estimation, and (3) graph-based clustering using the TrackletNet.

Trajectory generation. Given per-frame vehicle detection boxes, based on the camera motion and the appearance similarity between two consecutive frames, vehicle tracklets are generated through the intersection-over-union (IoU) with epipolar geometry constraint compensation. Each generated tracklet is regarded as a node when constructing a graph representation for tracklet generation.

Trajectory connectivity estimation. The weight of each graph edge spanning the two nodes (tracklets) in the graph measure the connectivity between two nodes, which represents the possibility of the two respective tracklets belonging to the same vehicle. Such connectivity are calculated considering physics and appearance models for linking tracklets [23].

TrackletNet graph-based clustering. The 4D positional information and 2048D appearance information of each tracklet are spread out in a 64D time dimension. These features are used in TrackleNet for calculating similarity scores. TrackleNet also helps associate tracklets in within each camera. Then, the method of [22] is used to perform clustering to minimize the total cost in the graph. After clustering, short tracklets with the same ID are combined to form longer tracklets in a consistent manner.

3.3. SCT Tracklet Post-Processing and Filtering

Tracklets produced by the TrackletNet tracker can still be broken or contain unwanted vehicles tracklets, in which many can be reliably recovered or ruled out using rule-based filtering. We first try to connect broken tracklets using the Re-ID features described in § 3.4. We next propose three simple but effective filters, namely *speed filtering*, *stay time filtering*, *IoU filtering*, to rule out undesired tracklets before feeding them for the consideration of CIR in § 3.5.

Re-ID appearance feature based tracklet linking. The id of tracklets may change if overlapping happens. Therefore, we connect the broken tracklets using appearance Re-ID features described in § 3.4. We calculated the pairwise cosine similarity of tracklets in a camera. The similarity between two tracklets is calculated with the last detection of the first tracklet and the first detection of another tracklet. Then, we can connect the tracklets by the similarity.

We observed three common problems of tracklets produced by the TrackletNet tracker, as shown in Fig. 3:

1. Tracklets without vehicle in the detected box that are with high, unstable "floating" speed, as in Fig. 3. Such high floating speed is due to the low confidence and erroneous detection and association of a false tracklet.
2. Tracklets without vehicle in the detected box, but with constant speed or being stationary, and appearing in either a very short or very long time span. Such false tracklets are caused by erroneous vehicle detection responses.
3. Tracklets arisen from the parked cars on the street, which are irrelevant and should not be accounted according to the challenge setup.

We developed three types of tracklet filtering to address the above three problems, which can rule out unsuitable tracklets in each camera view.

Tracklet speed filtering rules out unreliably tracklets with high "floating" speed by:

$$\frac{v - \mu_v}{\sigma_v} > \tau_v, \tag{1}$$

where vehicle velocity v is calculated using vehicle GPS positions and time stamps. The vehicle GPS coordinates can be calculated via the *planar homography* mapping of the image pixels to the GPS ground plane coordinates in terms of longitudes and latitudes, and the camera calibration homography matrix is provided by the AI City challenge organization. μ_v is the average speed of tracklets in one camera, and σ_v is the standard deviation of speed of tracklets, and both can be calculated from the training set. μ_v and σ_v

represent the distribution of vehicle speeds in each camera, and they are different for each camera. Threshold τ_v controls the aggressiveness of the deletion of tracklets under this rule. We empirically determine τ_v such that the amount of removed tracklets does not exceed 3% of the total number of tracklets.

Tracklet stay time filtering removes tracklets with irregularly staying time by:

$$\left| \frac{t - \mu_t}{\sigma_t} \right| > \tau_t, \tag{2}$$

where t denotes the during of a tracklet with fixed speed. We empirically found that, in most cases the "empty" boxes without a vehicle within it are highly unstable and appearing irregularly in a short or long period of time. This rule can effectively remove these erroneous tracklets.

Tracklet IoU filtering removes stationary vehicle tracklets by:

$$\frac{box_s \cap box_e}{box_s \cup box_e} > \tau_{iou}, \tag{3}$$

where box_s and box_e represent the first and last detection boxes of a tracklet under consideration. If the IOU of the first detection box and the last detection box is larger than a given threshold, it represents a stationary (parked) vehicle to be removed.

The removal of erroneous tracklets also significantly reduce the computational time of multi-camera tracking, as only suitable tracklets are to be considered in the later stage.

3.4. Vehicle Re-ID Feature Learning

We extract vehicle re-identification features using `ResNet-101` together with `IBN-Net-a`. The Instance-Batch Normalization Network (IBN-Net) [16] is a winning method from the Drivable Area Segmentation contest of the 2018 WAD Challenge, where the goal is to determine road that the vehicles drive on or regions they can potentially drive on. Unlike `ResNet` as an independent network, `IBN-Net` can be combined with other deep learning models to improve performance without increasing the computational cost. Here we combine `IBN-Net` with `ResNet-101` for Re-ID feature extraction. In this paper, we do not leverage information other than image features (such as vehicle license plate or make) for vehicle Re-ID.

IBN-Net can improve image appearance modeling via the combination of two normalization layers: instance normalization (IN) and batch normalization (BN). To reduce the feature changes introduced by superficial appearance without affecting the recognition of more profound content, we only add IN to the superficial level. To better preserve the image content in the superficial level, we replace half of the BN in the superficial level by IN.

The difference between IN and BN is that IN uses statistics of each sample to localize features. IN learned features should be less affected by appearance changes such as color, style, and virtuality (vs reality). In comparison, BN uses mini-batch with statistic mean and variance, and features in each channel are normalized during training. BN is preferred in the case when the focus is image content. Also, BN can speed up training and better learn distinguishing features. In summary, IBN-Net can better learn vehicle styles and image contents in the top layers.

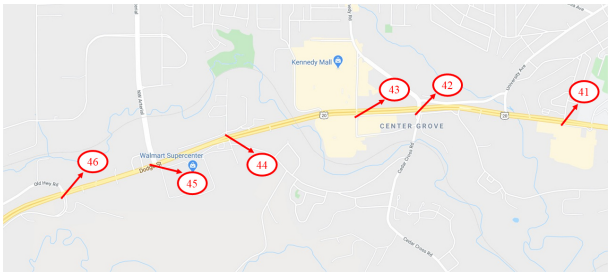


Figure 4. Camera locations in the AI City Challenge 2021 Track 3 MTMC test set.

3.5. Multi-target Tracking via CIR Association

Main objective of MTMC tracking is to perform vehicle tracklet association across cameras, which consists of the following steps.

Travel direction filtering. We infer the traveling direction of each vehicle using the GPS coordinates of the tracklets. The GPS coordinates are obtained from the provided homography calibration as discussed in § 3.3. We calculate the cosine of the angle between two tracklets to determine whether they are travelling in the same direction or not. If the cosine angle < 0.5, the two vehicles are travelling in different directions, and thus we do not considering linking such tracklets together. Since all test cameras in the AI City 2021 challenge are deployed at a single road as in Fig. 4, enforcing such travel direction filtering is very effective for MTMC tracklet association.

Tracklet match pre-ranking. For the remaining tracklets, we consider all possible pairwise matching across camera views. Due to the large number of tracklets to consider, for a target tracklet A in a camera view, consider all possible candidate tracklets in another view ranked ordered by the Re-ID similarity scores calculated using the Re-ID features from § 3.4. For a vehicle tracklet with multiple detections, we calculate the mean and standard deviation of vehicle detection feature at each frame to be used for cosine similarity comparisons. We first consider candidate tracklets with higher similarities according to:

$$\frac{sim - \mu_{sim}}{\sigma_{sim}} >= 1. \tag{4}$$

where sim denotes the similarity between a tracklet A and its candidate tracklets in the matching list L_A. μ_{sim} de-

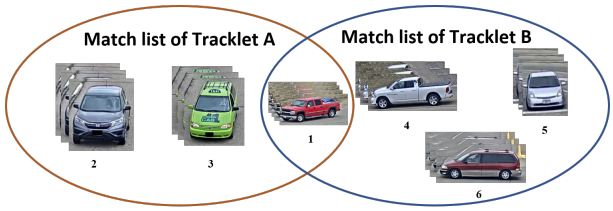


Figure 5. A CIR tracklet association example in the proposed MTMC vehicle tracking. Here we consider the association of two tracklet matching lists L_A and L_B, where L_A contains three tracklets $\{1, 2, 3\}$ and and L_B contains four tracklets $\{1, 4, 5, 6\}$. The intersection of L_A and L_B is a single vehicle tracklet $\{1\}$. According to Eq. (6), $CIR(L_A, L_B) = \frac{1}{min(3,4)} = \frac{1}{3}$.

notes the mean of similarities of the matching list L_A, and σ_{sim} denotes the standard deviation of similarities of the matching list L_A. We can next calculate the ratio of chosen tracklets to the total amount of possible tracklets by:

$$R_s = \frac{N_c}{N_{total}} \tag{5}$$

where N_c represents the amount of the candidates in the matching list L_A, and N_{total} represents the total amount of tracklets in the matching list L_A. If R_s is bigger than 15%, we consider that the tracklet A does not matching any tracklet. In this case, σ_{sim} is too small so that the similarity between query tracklet and other tracklets is very close.

Candidate Intersection Ratio (CIR) tracklet matching. We develop the CIR tracklet matching metric to evaluate the similarity between two tracklets for the calculation and ranking of tracklet associations iteratively, and the association will be performed hierarchically similar to the standard **agglomerative clustering** algorithm in *hierarchical clustering*. We follow a similar notation in considering a tracklet A with matching candidates L_A and another tracklet B with matching candidates L_B. The CIR metric for evaluating the association similarity of matching list of tracklets L_A and L_B is defined as:

$$CIR(L_A, L_B) = \frac{size(L_A \cap L_B)}{min(size(L_A), size(L_B))} \tag{6}$$

Fig. 5 shows an example of the CIR metric calculation, where the intersection of the matching list of two tracklets are calculated according to Eq. (6) to determine if the two tracklets should be associated together (*i.e.* merged or linked, and regarded as the tracklets of the same vehicle across views).

The **CIR tracklet association algorithm** performs iteratively by considering all such (L_A, L_B) pairs across views. Initially, all distinct (un-associated) tracklets are regarded as individual nodes without edges connecting them. The matching list of associated tracklets are constructed by associating a the pair (L_A, L_B), and repeated association yields

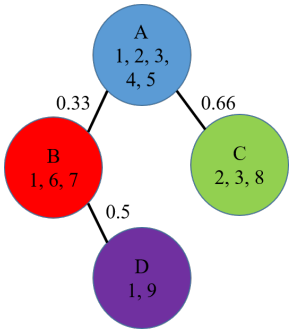


Figure 6. **A tracklet association example during the execution of the CIR association algorithm.** Suppose tracklet B has the max CIR with tracklet A. Due to size of tracklet B is smaller than size of tracklet A, Tracklet A is assigned as tracklet B's parent. We repeat this process and group the tracklets as a tree.

a binary tree-like structure, where the root of the tree indicates the vehicle ID of this association tree. This way, the association tree is constructed by regarding tracklets and their CIR similarity as nodes and edge weight. During tree construction, we enforce that the size of parent node must be larger or equal to the size of the children nodes. This way an unique vehicle ID should be retained for each tree throughout the whole iterative association process.

We iteratively compute the CIR score between every tracklet and the tracklets of sizes smaller or equal to the first one. Fig. 6 illustrate an example of the tracklet association. In the association of tracklet A to its candidates, we find that tracklet B contains larger CIR score with tracklet A (where the score $>$ threshold). Therefore, tracklet A is assign as the parent node of tracklet B. By repeating this process, we can cluster the same tracklets as a tree.

In contrast to a naive greedy best-first tracklet association algorithm that only consider the association of the best matching pair of tracklets, the proposed CIR tracklet association consider the matching of the whole matching list of tracklets for optimization.

Avoid cyclic tracklet associations. Repeated tracklet association following the above steps by associating tracklets with best-first selection of CIR scores might results in an unwanted case of *cycle associations*, where the resulting vehicle ID of the associated set cannot be uniquely defined and causing problematic MTMC results. To avoid such issue, we enforce the following rules during the CIR association steps to explicitly check and avoid cyclic associations.

1. When the size of the matching list of tracklet A is larger than size of the matching list of tracklet B, we assign tracklet B as the child node of tracklet A.

2. If the size of the matching list of tracklet A is equal to size of the matching list of tracklet B, we check and make sure tracklet B is not the ancestor of A. If track-

let B is indeed the ancestor of A, we assign tracklet B to be the parent of tracklet A. Otherwise, we assign tracklet A to be the parent of tracklet B.

The CIR association iteration terminations when all pair-wise tracklet lists are examined. Afterwards, we assign the vehicle ID of each tree root to all of its descent nodes. This completes the CIR multi-camera vehicle tracking. Algorithm 1 shows the detailed steps of the CIR vehicle tracklet association algorithm in pseudo code.

Algorithm 1: CIR Tracklet Association

Input: tracklets $T_1, ..., T_N$, number of tracklets N
Output: tracklets after association $T'_1, ..., T'_N$
for *i=1 to N* **do**
  Calculate the pair-wise cosine similarity $S_{i,j}$
  Obtain $T_i.L$ the matching list of T_i using $S_{i,j}$
  $T_i.L := L$
  //initialize CIR threshold
  $T_i.max\_score := 0.33$
for *i=1 to N* **do**
  for *j=1 to N* **do**
    if *(i==j) or* ($T_i.camera == T_j.camera$) *or* (*size*(T_i) < *size*(T_j)) **then**
      continue
    $score := CIR(T_i.L, T_j.L)$
    if *size*(T_i) == *size*(T_j) **then**
      if $is\_ancestor(T_i, T_j)$ **then**
        if $score > T_i.max\_score$ **then**
          $T_i.parent := T_j$
          $T_i.max\_score := score$
      else
        if $score > T_j.max\_score$ **then**
          $T_j.parent := T_i$
          $T_j.max\_score := score$
    else
      if $score > T_j.max\_score$ **then**
        $T_j.parent := T_i$
        $T_j.max\_score := score$
for *i=1 to N* **do**
  $T'_i.id := T_i.root\_id$

4. Experimental Results

Dataset. The AI City Challenge 2021 organization provides 3.58 hours of traffic videos collected from 46 highway and street cameras spanning 16 intersections in a mid-sized U.S. city. There are 58 videos recorded by multiple cameras in the training and verification sets, and there are 6 test videos. Baseline results include the detection, vehicle re-identification, and single camera tracking results are also provided.

Evaluation Metrics. Given the true-positive TP_{id}, false-positive FP_{id}, and false-negative FN_{id} of the detection and groundtruth vehicle IDs, multi-camera vehicle tracking performance is evaluated using the F1 score of vehicle identity (IDF1):

$$IDF1 = \frac{2TP_{id}}{2TP_{id} + FP_{id} + FN_{id}}. \quad (7)$$

The proposed CIR MTMC tracking obtained IDF1 score of 0.1343. It ranks the 18-th out of 20 total submissions from participant teams on public leaderboard of this Track 3 Challenge.

Method	IDF1
Hungarian	60.22%
Tracklet Clustering	61.00%
TrackletNet Tracker	69.31%

Table 1. Comparison of a baseline Hungarian tracking and TNT tracker on the challenge validation set.

filters	IDF1
baseline	40.99%
speed	44.10%
speed + staytime	51.41%
speed + staytime + IOU	61%

Table 2. **SCT Tracklet Filtering Results.** The baseline denotes the TNT tracking results with only the removal of overlapping boxes.

4.1. Ablation Study on TNT for SCT

We provide performance evaluation of the TrackletNet Tracker (TNT) in § 3.2 for single-camera tracking. Since the AI City Challenge 2021 organization did not provide the TNT tracking results on the challenge validation set, we must train our own TNT model from scratch. We use initial learning of 0.001 for the TNT training. We gradually reduce learning rate by ten times for every 2000 steps until the learning rate reaches 0.00001 and we stop training at that time.

We compare our trained TNT tracker with the baselines of (1) Hungarian algorithm and (2) Tracklet Clustering on the challenge validation set. Results are shown in Table 1. The performance of TNT increases by 9% when compared to the Hungarian algorithm.

4.2. Ablation Study on Post SCT Tracklet Filtering

Table 2 shows the experimental evaluations of the three SCT tracklet filtering methods described in § 3.3. Significant performance improvements of almost 20% increase of IDF1 score are obtained by incorporating all three filters. Despite their simplicity, empirical validations of these three filters show that they can indeed address many com-

Figure 7. **Vehicle Re-ID results.** The query image in on the left hand side. The top 50 gallery images that match the query image are on the right hand side.

mon problems that can occur in the real-world setting for MTMC vehicle tracking.

4.3. Ablation Study on Vehicle Re-ID

We evaluate the vehicle Re-ID performance using different combination of backbone models. Results show that the normalization design of `IBN-Net` described in § 3.4 brings significant improvement.

Experimental settings. Input to the ablation study Re-ID models are 224×224 images. Data augmentation methods include random horizontal flip, padding, and random erasing. During training, we use the aggregation of cross-entropy loss and triplet loss. STD optimizer is used with initial learning rate 0.01, momentum 0.9, and weight decay 0.0005. In addition to the optimizer, we also use the WarmupMultiStepLR as scheduler. In the first ten epochs, we increase learning rate linearly from 0.001 to 0.01, which can ease the initial training instability. In the 40-th epoch and the 70-th epoch, we reduce the learning rate to 0.001 and 0.0001, respectively, which helps model convergence in the later training stages. The training ends at 100 epochs.

Fig. 7 shows an example result of vehicle Re-ID. The Re-ID model retains a strong discriminative features for vehicles that may come from different camera with distinct view points and viewing orientations.

Table 3 shows the ablation study results with various combinations of backbone models. The combination of `ResNet101` and `IBN-Net-a` performs the best on the challenge validation set, with 30.49% mAP. The `IBN-Net-a` better improves feature learning thanks to its normalization design. It contributes about 10% of mAP performance improvement.

4.4. Ablation Study on MTMC Tracking

We compare CIR tracklet association with a baseline method based on simple matching with top-1 cosine similarity. Table 4 shows the comparison results. Observe that the IDF1 score increases by almost 2.5% in the proposed CIR approach. The CIR improvement is mainly on the decreasing of false positives by almost 50% of the baseline.

Model	mAP
`ResNet50` + `ibn-a`	27.41%
`ResNet50` + `ibn-a` + data aug.	27.2%
`ResNet101` + `ibn-a`	29.09%
`ResNet101` + `ibn-a` + data aug.	30.49%

Table 3. Re-identification mAP results on the challenge validation set using various backbone models and data augmentation.

Method	IDF1	IDTP	IDFP	IDFN
Top-1	32.65%	48718	64305	136709
CIR	35.13%	47094	35610	138333

Table 4. Comparison of CIR tracklet association against the top-1 baseline on the challenge validation set.

This is intuitive, as cosine similarity is not always suitable to compare and rank visually similar vehicles that can appear in different viewing angles or orientations under different cameras. The proposed CIR metric can essentially provide a more robust and "soft" ranking when evaluating the merging of two set of candidate tracklets by considering the whole matching list of tracklets. CIR can thus tolerate misleading information from the cosine similarity to avoid erroneous matching.

5. Conclusion

We presented a new robust multi-target, multi-camera (MTMC) vehicle tracklet association method based on a new Candidate Intersection Ratio (CIR) tracklet association approach. The proposed CIR tracklet association algorithm are capable to perform large-scale, off-line MTMC vehicle tracking. Results are submitted to the AI City 2021 Challenge on Track 3 MTMC tracking contest, where scores are compared against other participant teams on the 2021 challenge leaderboard. The three SCT tracklet filtering rules based on vehicle tracklet speed, stay time, and IoU filtering also large improve the MTMC IDF1 score. The CIR algorithm performs tracklet association by considering all candidate tracklets on a tree-like hierarchical clustering data structure. The robustness in such design can outperform baseline association methods relying on merging tracklets with top-1 similarities, while reducing false-positives and false-negatives.

Future Works include the investigation of: (1) better vehicle tracklet association metric that might outperform the cosine similarity, (2) adopting the proposed method to handle on-line streaming of multiple traffic videos, as well as (3) performance optimization to run the pipeline on edge devices and embedded platforms.

References

[1] Alexey Bochkovskiy, Chien-Yao Wang, and Hong-Yuan Mark Liao. YOLOv4: Optimal speed and accuracy of object detection. In *arXiv 2004.10934*, 2020. 2

[2] Martin Danelljan, Goutam Bhat, Fahad Shahbaz Khan, and Michael Felsberg. ECO: Efficient convolution operators for tracking. In *arXiv 1611.09224*, 2017. 2

[3] M. Danelljan, G. Häger, F. S. Khan, and M. Felsberg. Learning spatially regularized correlation filters for visual tracking. In *International Conference on Computer Vision (ICCV)*, pages 4310–4318, 2015. 2

[4] A. Geiger, M. Lauer, C. Wojek, C. Stiller, and R. Urtasun. 3d traffic scene understanding from movable platforms. *IEEE Trans. Pattern Analysis and Machine Intelligence (PAMI)*, 36(5):1012–1025, 2014. 2

[5] Yiluan Guo and Ngai-Man Cheung. Efficient and deep person re-identification using multi-level similarity. In *Computer Vision and Pattern Recognition (CVPR)*, June 2018. 2

[6] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross B. Girshick. Mask R-CNN. *CoRR*, abs/1703.06870, 2017. 3

[7] Shuting He, Hao Luo, Weihua Chen, Miao Zhang, Yuqi Zhang, Fan Wang, Hao Li, and Wei Jiang. Multi-domain learning and identity mining for vehicle re-identification. In *Proc. CVPR Workshops*, 2020. 2

[8] J. F. Henriques, R. Caseiro, P. Martins, and J. Batista. High-speed tracking with kernelized correlation filters. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 37(3):583–596, 2015. 2

[9] Hung-Min Hsu, Tsung-Wei Huang, Gaoang Wang, Jiarui Cai, Zhichao Lei, and Jenq-Neng Hwang. Multi-camera tracking of vehicles based on deep features re-id and trajectory-based camera link models. In *Computer Vision and Pattern Recognition (CVPR) Workshop*, June 2019. 1

[10] Licheng Jiao, Fan Zhang, Fang Liu, Shuyuan Yang, Lingling Li, Zhixi Feng, and Rong Qu. A survey of deep learning-based object detection, 2019. 2

[11] Young-Gun Lee, J. Hwang, and Zhijun Fang. Combined estimation of camera link models for human tracking across nonoverlapping cameras. *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 2254–2258, 2015. 3

[12] Hongye Liu, Yonghong Tian, Yaowei Yang, Lu Pang, and Tiejun Huang. Deep relative distance learning: Tell the difference between similar vehicles. In *Computer Vision and Pattern Recognition (CVPR)*, June 2016. 2

[13] Wei Liu, Dragomir Anguelov, Dumitru Erhan, Christian Szegedy, Scott Reed, Cheng-Yang Fu, and Alexander C. Berg. SSD: Single shot multibox detector. In *European Conference on Computer Vision (ECCV)*, 2016. 2, 3

[14] Wenqian Liu, Octavia Camps, and Mario Sznaier. Multi-camera multi-object tracking. In *arXiv 1709.07065*, 2017. 3

[15] Milind Naphade, Shuo Wang, David C. Anastasiu, Zheng Tang, Ming-Ching Chang, Xiaodong Yang, Liang Zheng, Anuj Sharma, Rama Chellappa, and Pranamesh Chakraborty. The 4th AI City Challenge. In *CVPR Workshop on AI City Challenge*, 2020. 1

[16] Xingang Pan, Ping Luo, Jianping Shi, and Xiaoou Tang. Two at once: Enhancing learning and generalization capacities via IBN-Net. In Vittorio Ferrari, Martial Hebert, Cristian Sminchisescu, and Yair Weiss, editors, *European Conference on Computer Vision (ECCV)*, pages 484–500, 2018. 5

[17] Y. Qian, L. Yu, W. Liu, and A. G. Hauptmann. ELECTRICITY: An efficient multi-camera vehicle tracking system for intelligent city. In *Computer Vision and Pattern Recognition (CVPR) Workshop*, pages 2511–2519, 2020. 1

[18] Joseph Redmon and Ali Farhadi. Yolov3: An incremental improvement. *CoRR*, abs/1804.02767, 2018. 3

[19] Shaoqing Ren, Kaiming He, Ross B. Girshick, and Jian Sun. Faster R-CNN: Towards real-time object detection with region proposal networks. In *NIPS*, pages 91–99, 2015. 1, 2, 3

[20] Siyu Tang, Mykhaylo Andriluka, Bjoern Andres, and Bernt Schiele. Multiple people tracking by lifted multicut and person re-identification. In *Computer Vision and Pattern Recognition (CVPR)*, July 2017. 2

[21] Zheng Tang, Milind Naphade, Ming-Yu Liu, Xiaodong Yang, Stan Birchfield, Shuo Wang, Ratnesh Kumar, David C. Anastasiu, and Jenq-Neng Hwang. CityFlow: A city-scale benchmark for multi-target multi-camera vehicle tracking and re-identification. In *Computer Vision and Pattern Recognition (CVPR)*, pages 8797–8806, 2019. 3

[22] Zheng Tang, Gaoang Wang, Hao Xiao, Aotian Zheng, and Jenq-Neng Hwang. Single-camera and inter-camera vehicle tracking and 3D speed estimation based on fusion of visual and semantic features. In *Computer Vision and Pattern Recognition (CVPR) Workshop*, June 2018. 2, 3, 4

[23] Gaoang Wang, Yizhou Wang, Haotian Zhang, Renshu Gu, and Jenq-Neng Hwang. Exploit the connectivity: Multi-object tracking with TrackletNet. In *ACM International Conference on Multimedia*, page 482–490, 2019. 3, 4

[24] Peng Wang and Qiang Ji. Robust face tracking via collaboration of generic and specific models. *IEEE Trans. Image Process.*, 17(7):1189–1199, 2008. 3