Spatio-temporal Consistency and Hierarchical Matching for Multi-Target Multi-Camera Vehicle Tracking

Peilun Li*, Guozhen Li*, Zhangxi Yan*, Youzeng Li*, Meiqi Lu*, Pengfei Xu*, Yang Gu*, Bing Bai* and Yifei Zhang*

*DiDi Chuxing
[1]lipeilun,liguozhen,xupengfei@didiglobal.com

Abstract

Recently, many approaches have been addressed to realize Multi-Target Multi-Camera(MTMC) vehicle tracking, which is critical in intelligent transportation system (ITS). Continuous improvements of MTMC have been limited by two modules - trajectory feature representation and feature metric in the city-scale camera condition. In this paper, we propose a spatio-temporal consistency and hierarchical matching method to overcome the challenges. As first step, a popular object detection and object tracking method are implemented to detect vehicles and track them in single camera, thus achieved high performance. The smoothness of trajectory and slice direction of movement make spatio-temporal consistency more confident. As second step, a bottom-up hierarchical match strategy is used to match targets in different cameras. Top performance in City-Scale Multi-Camera Vehicle Tracking task at the NVIDIA AI City Challenge 2019 demonstrated the advantage of our methods.

1. Introduction

In recent years, the research on intelligent transportation system(ITS) has attracted much attention in academia and industry. Video data collected by fixed multi-camera are of great significance for traffic feature estimation, traffic anomaly detection, multi-camera tracking and other applications. General object recognition/detection/tracking/re-identification(ReID) has been extensively studied in the past. But the image features of vehicles in traffic environment are obviously different from those of general objects, so we can not directly reuse the algorithm modeled for general object to the vehicles in traffic video scenario. The difficulty of vehicle tracking is that the similarity between different vehicles is very small, so sometimes the same vehicle under different perspectives of the image will be very distinctive. High traffic density is and the serious occlusion phenomenon bring great challenges to the multi-target and cross-camera vehicles tracking.

In the past, a large number of excellent multi-target single camera tracking(SCT) algorithms have emerged, however, the single-camera tracking of vehicles is still very challenging, because many factors seriously interfere with the performance of the tracking algorithm including real traffic conditions, vehicle, aspect ratio, the image of the scale of the local distortion, illumination changes and noise interference and image fuzzy, and the shade between vehicles and so on. Although the literature proposed a solution to ease the occlusion, it still needs to develop robust single camera tracking algorithm to accurately capture the vehicle.

Multi-camera tracking(MCT), sometimes interpreted as re-identification, has been extensively studied in recent years in the field of pedestrian tracking. The main idea is to use the deep neural network as the feature extractor to calculate the metric of the features and optimize the entire ReID model pipeline, so as to achieve the state-of-the-art performance. Pedestrian image is relatively simple compared with the characteristics of the vehicle image, and the vehicle under different perspective to observe with completely different image characteristics, it brought great difficulties to the vehicle ReID task, the scarce research on vehicle ReID is a embarrassing situation, how to build a robust vehicle ReID model for the whole process of MTMC used is also very important. Besides, GPS information is significant complement to MCT result, how to combine video information with GPS location is a problem remain unsolved. The organization of our paper is described as follows: Section 2 introduces several related works about single-camera tracking and vehicle ReID, Section 3 describes our whole pipeline, Section 4 shows our experimental results and we conclude our paper in Section 5.

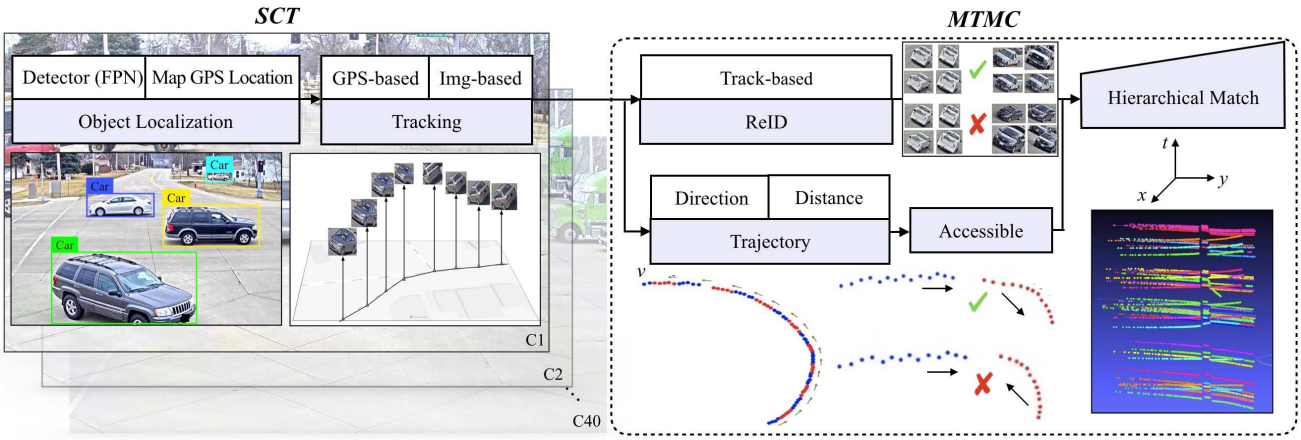

Figure 1. Our whole pipeline consists of two parts: SCT and MTMC. Two critical methods in SCT function realization are Object Localization and Tracking. In Object Localization, two algorithms, which are FPN detector and mapping from bounding box to GPS location, are combined. GPS-based feature, as output of Object Localization, combined with image-based feature achieved Tracking. MTMC module hierarchically matched Tracks from SCT, using fusion of ReID feature, spatio-temporal feature and GPS-trajectory feature.

2. Related Works

2.1. Detection

Object detector is usually based on proposal. R-CNN[5] is among the first CNN-based detection approach, which generates proposal by selective search method [19]. Then Fast R-CNN [4] is introduced to enable CNN feature map to be shared in network. Faster R-CNN [16] further introduce a Region Proposal Network(RPN) to improve proposal quality, which greatly improves the detection performance. Another option is one stage detection network, which predicts the bounding boxes without proposal generation, so it can be more quickly. YOLO [14] and YOLOv2[15] regard object detection as a regression problem, which split image into multiple gird cells and predict the bounding boxes and associated class probabilities in each grid cell. SSD[11] predicts the bounding boxes of different scales in different layers, which improves one stage detector performance in complex scene. In conclusion, two stage (proposal-based) detector focus on high accuracy while one stage detector has advantage in speed.

2.2. Single-camera tracking (SCT)

The main challenge of Single camera tracking is object occlusion, recording frame loss and multi-target matching between frames. In recent years, there have been two main methods widely used for multi-target tracking. One is to convert the MOT into a data association problem, such as,Joint Probabilistic Data Association (JPDA)[3]. The other is using Deep Neural Network (DNN) that simultaneously models association among indefinite number of objects. Heng et al. give a benchmark in large-scale single object tracking[2]. Yao[23] et. provide a comprehensive review of the state-of-the-art tracking methods, and classify these methods into different categories, and identify new trends.In order to compare diverse object tracking optimization strategies, Xu. et. use the Expectation Over Transformation (EOT) algorithm to generate physical adversaries that fool tracking models[21].

2.3. Multi-camera multi-object tracking (MTMC)

With the development of urban intelligence, using traffic cameras as sensors have greatly improved the study of multi-target multi-camera (MTMC) tracking[[17]]. Liu et. propose a pipeline for multi-target visual tracking under multi-camera system which extracts both appearance and dynamic motion similarities[12]. To tackle loss of tracking target, Wu et. propose a generic multiview tracking (GMT) framework which the key point is a cross-camera trajectory prediction network (TPN)[20].

2.4. Vehicle Re-Identification

Deep neural networks(DNN) benefits greatly to re-identification. In recent years, many promising results have been achieved in Person Re-identification. But few works for car Re-Identifications. Zhtang et al[18]. extend their work and fuse deep learning features, detected license plate features and detected car types, for vehicle re-identification which is selected as the winning method in NVIDIA AI City Challenge 2017. In [13][1], license-based recognition and comparison can be used in vehicle re-identification, But low quality image capturing, low resolution license plate, fake

license plate may make re-identification failure.

3. Methodology

First, two concepts need to be described, track and GPS-trajectory. Each track contrains multiple images of the same vehicle captured by one camera, while GPS-trajectory only contains GPS coordinates and timestamp of one track.

3.1. Vehicle Detection based on FPN

Feature Pyramid Networks (FPN)[10] uses inherent multi-scale, pyramidal hierarchy of deep convolutional networks to generate high quality feature map by fusing low-level features with high-level semantic feature maps and high-level features with rich location information, which achieves the best performance in our task scene.

3.2. Single-Camera Multi-Vehicle Tracking

3.2.1 Tracking algorithm

Our tracking algorithm is based on ReId feature, spatio-temporal feature and GPS information. For bounding box pair $\langle i, j \rangle$ in adjacent frames, we define a tracking loss function as

$$loss_{tracking}(i,j) = \| r_i - r_j \|_2 + \lambda_g \| g_i - g_j \|_2 \quad (1)$$

where r_i and r_j are ReID feature vector with 2048 dimensions, g_i and g_j are GPS coordinates with 2 dimensions, λ_g is set as 0.05. Hungarian Algorithm[9] is used to find bounding box pair with minimum loss function value.

3.2.2 Post processing

Lost target in tracking is widespread due to occlusion, to solve this issue, the matching strategy is determined by two different thresholds used in short and long term occlusion respectively. And the thresholds are set based on the minimum distance between tracks, which is defined as the minimum ReID feature distance between the last three boxes of the former track and the first three boxes of the latter track.

3.3. Vehicle Re-Identification

We use vehicle re-identificaion(ReID) model to achieve multi-camera object tracking. A robust ReID model requires training data of high quality. In order to obtain promising performance in multi-camera tracking, we use AI City Challenge dataset as our training data.

The dataset includes 333 different vehicle identities, we split them to training set and validation set, among which the training set contain 149 vehicle identities and the validation set contains 184 vehicle identities. The performance of the model on the validation set is used to measure the performance of ReID model. During model reasoning, video frame is extracted and cropped, then fed to ReID model for inference.

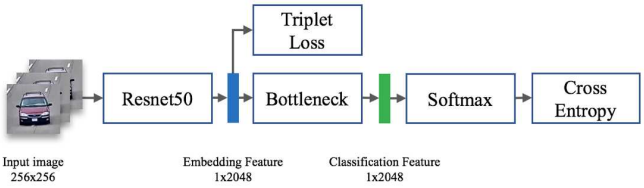

Figure 2. ReID model pipeline

The pipeline of the whole ReID model consists of two parts, Resnet50[7] backbone network as the feature extractor and the loss function designed for ReID task. In the pre-trained Resnet50 model, we follow the work in [22] which effectively improves our ReID performance by introducing the bottleneck structure. The loss function includes two parts: the cross entropy loss for classification and the triplet loss for metric learning. Since the training set consists of 149 vehicle identities, the output of the model is the softmax classifier of dimension 149, for unseen vehicle identities, we need features to embbed the same vehicle identities into same space for different images, we introduce triplet loss[8] to reduce the distance between the images in the same class and enlarge the distance between the images in different classes. In the process of model training, we use the warmup strategy [6], so that the model can converge to a better solution.

We pad the input cropped image by 10 pixels in border replicate method, and resize it to 256×256, and feed it to ReID model. Adam optimizer with a learning rate of 0.00035 is adopted to solve the model parameters. Random horizontal flip, an augmented operation, is used in the training process to obtain more training data. In the sampling process, we set batchsize $N = 64$, and each batch contained 4 vehicle identities. Triplet loss was calculated in each batch for model optimization represented in formula, x_i^a is the anchor sample, x_i^n is the farthest sample in the same class with x_i^a and x_i^p is the nearest sample in different classes with x_i^a. The whole loss function is formulated as

$$l_t = \sum_{i=1}^{N} [||f(x_i^a) - f(x_i^p)||_2^2 - ||f(x_i^a) - f(x_i^n)||_2^2 + \alpha]_+ \quad (2)$$

$$l_c = -\sum_{i=1}^{N} [\sum_{j=1}^{n} y_j log(x_{ij})] \quad (3)$$

$$loss(x_i, y_i) = l_t + \lambda \times l_c \quad (4)$$

We simply set $\lambda = 1$ here. Finally we get a promising vehicle ReID model.

After obtaining all the features in the cropped image of all single tracks, we use minimum distance value as the met-

ric of two distinct tracks. We will show the ablation experiment result for optimal calculation method for distance metric in section 4.2.

3.4. GPS-trajectory feature extraction

Based on detection and tracking results, we obtained a large number of trajectories, each belonging to the same vehicle. The trajectory and point in it can be represented as

$$p_i = (id, lon_i, lat_i, t_i) \tag{5}$$

$$Traj = \{p_1, p_2, p_3, \ldots, p_n\} \tag{6}$$

where id identifies the trajectory, t_i is timestamp of each point p_i, lon_i and lat_i are the longitude and latitude of the vehicle in t_i that we obtained from a transform matrix M and coordinates in images. In addition, for a vehicle with bounding box $b = (x, y, w, h)$, we use point $c = (x + \frac{1}{2}w, y + \frac{4}{5}h)$ to calculate its GPS coordinate, which can be calculated by

$$(lon_i, lat_i, h_i) = M \cdot (c_x, c_y, 1)^{\mathrm{T}} \tag{7}$$

Besides appearance feature, trajectory also has temporal feature and 2D spatial feature. An innovative strategy is employed to improve the matching accuracy and efficiency. Firstly, we smooth the trajectory to remove outliers that move too fast, which introduced by detection error or timestamp error. Secondly, we fully extract features of trajectory in the following three aspects: direction of movement, minimum distance estimation and trajectory accessibility.

3.4.1 Trajectory smoothing

Since the speed of vehicles cannot change abruptly, we believe that there should be stability of speed between adjacent points. For all trajectories that contain more than three points, we compare the speed among p_i, p_{i-1}, p_{i-2} with λ_{speed} to judge the point is outlier or not.Motion vector is denoted as $\vec{v}_{i,i-1} = (lon_i - lon_{i-1}, lat_i - lat_{i-1})$, and the speed is calculated as

$$s_{i,i-1} = \frac{\parallel \vec{v}_{i,i-1} \parallel_2}{t_i - t_{i-1}} \tag{8}$$

There are three propositions to represent judgements, $Q : s_{i,i-1} > \lambda_{speed}$, $R : s_{i-1,i-2} > \lambda_{speed}$, $T : s_{i,i-2} > \lambda_{speed}$. Hence,we can judge p_i is outlier by following normal form

$$"p_i \; is \; a \; outlier" \Leftrightarrow Q \vee (\neg Q \wedge R \wedge T) \tag{9}$$

λ_{speed} is set as 80 in our experiments. Our outlier deletion strategy is conservative in order to minimize the influence of wrong deletion on trajectory matching.

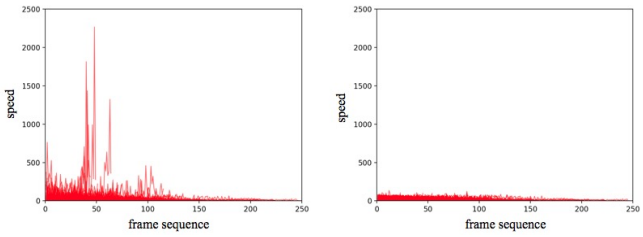

Figure 3. The left line chart is speed before trajectory smoothing, the right one is after smoothing.

3.4.2 Direction of movement

After we removed outliers, a roughly smooth trajectory can be obtained, and the idea of slicing is introduced. For one trajectory consists of s slices whose length is l, the average value point of the j-th slice can be defined as

$$p_j = (\frac{1}{l}\sum\nolimits_{i=j*l-l+1}^{j*l} lon_i, \frac{1}{l}\sum\nolimits_{i=j*l-l+1}^{j*l} lat_i) \tag{10}$$

where $j = 1, 2, \ldots, s$ and then the motion vector of average value point in adjacent slices is calculated as $\vec{v}_{j+1,j} = (lon_{j+1} - lon_j, lat_{j+1} - lat_j)$, the sequence of motion vector is $V = \{\vec{v}_{j+1,j}\}, j = 1, 2, \ldots, s-1$. For $Traj_p$ and $Traj_q$ containing s_p and s_q slices respectively, whose motion vector sequences are $V_p = \{\vec{v}_{i+1,i}\}, i = 1, 2, \ldots, s_p - 1$ and $V_q = \{\vec{v}_{j+1,j}\}, j = 1, 2, \ldots, s_q - 1$, the similarity sequence of two trajectories can be obtained by computing cosine similarity in pairs

$$sim_{p,q} = \left\{ \frac{\vec{v_p}^T \vec{v_q}}{\parallel \vec{v_p} \parallel_2 \parallel \vec{v_q} \parallel_2} \right\} \tag{11}$$

where $\vec{v_p} \in V_p, \vec{v_q} \in V_q$. $sim_{p,q}$ represents the similarity of two trajectories. The length of slice l and λ_{sim} are empirically set as 10 and -0.5 to judge direction consistency.

3.4.3 Minimum distance estimation

With the same slicing idea, the minimum distance between $Traj_p$ and $Traj_q$ can be estimated as the minimum distance between their average value points of each slice, $p_p = (lon_p, lat_p)$ and $p_q = (lon_q, lat_q)$. The distance sequence is given by

$$D_{p,q} = \{\parallel (lon_p - lon_q, lat_p - lat_q \parallel_2\} \tag{12}$$

We believe that the matching probability is very limited if the minimum distance between two trajectories is large enough.

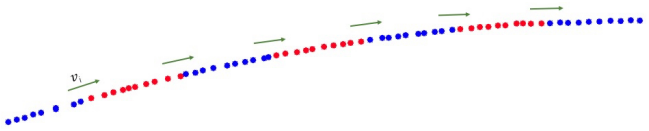

Figure 4. Trajectory slicing. The red dots and blue dots represent points belong to adjacent slices, and the green arrow is motion vector between adjacent slices.

3.4.4 Trajectory accessibility

For $Traj_p$ and $Traj_q$, the accessibility that $Traj_p$ can access $Traj_q$ is calculated as following algorithm 1.

Algorithm 1 Accessibility calculation

Input: Dis_{min}, Sim_{max}, v_{p_0,q_0}, v_{mean}, S
Dis_{min} and Sim_{max} are minimum distance and maximum similarity between two trajectories.
v_{p_0,q_0} is the motion vector between start points.
v_{mean} is the average motion vector of one trajectory.
S is function to compute similarity between vectors.
Output: σ, the accessibility from one trajectory to the other one.

σ = 1
if $Dis_{min} > \lambda_{dis}$ **then**
  σ = 0
else if $Sim_{max} < \lambda_{sim}$ **or** $S(V_{p_0,q_0}, V_{mean}) < \lambda_{sim}$ **then**
  σ = -1
end if
return σ

σ is 1 represents there is probability one trajectory can access the other one, and -1 represents there is not, and 0 indicates two trajectories may have some overlapping parts. λ_{dis} is set to 10 in our experiments.

In this part, we smooth the GPS-trajectory, calculate the movement direction similarity, minimum distance and accessibility between trajectorie pairs. Our strategy keeps the results as raw as possible and provides a large index table to Multi-Target Multi-Camera Tracking.

3.5. Multi-Target Multi-Camera Tracking

To improve the accuracy of track matching, we use three stages operation:

1. Matching tracks based on ReID features.

2. Finetuning Matching tracks based on multi-source information

3. Post-processing for remaining single track.

3.5.1 Basic Components

Time IoU Two adjacent tracks of the same car usually have similarities in time. Here we use IoU of two time periods to judge the similarity of time.

$$IOU_{time} = \frac{t_1 \cap t_2}{t_1 \cup t_2} \tag{13}$$

Due to noise in video transmission, which is common in real deployed systems, some frames are skipped within some videos, so reasonable time fluctuations are allowed. For time period $t = (s_0, s_1)$,we broadened the period to enhance his fault tolerance $t_+ = (s_0 - \delta, s_1 + \delta)$.

$$IOU_{time_+} = \frac{t_{1_+} \cap t_2}{t_{1_+} \cup t_2} \tag{14}$$

Track Direction Similarity In 3.4 section, we defined how to calculate the cos value of the angle between two trajectories. Here, when the angle between two trajectories is greater than the threshold th, we consider that the two trajectories are in the same direction.

3.5.2 Multi-Target Multi-Camera Tracking

Algorithm 2 Track matching

Input: $AllTracks$, th_1, th_2, th_3,D, C, R
$AllTracks$ is all tracks.
$thres_{sim}$ is threshold for deep feature similarity.
D, C and R are functions judging movement directions and camera id consistency, deep feature similarity of two tracks.
Output: $OutputTracks$, the fused tracks.

for $round$ in (1,2,3) **do**
  for $\langle Track_p, Track_q \rangle$ in $AllTracks$ **do**
    d = D($Track_p, Track_q$)
    c = C($Track_p, Track_q$)
    r = R($Track_p, Track_q$)
    if d is True **and** c is False **and** $r < thres_{sim}$ **then**
      $OutputTracks$.append($Track_p + Track_q$)
    else
      $OutputTracks$.append($Track_p$)
      $OutputTracks$.append($Track_q$)
    end if
  end for
end for

return $OutputTracks$

stage 1 matching tracks based on ReID features In this stage, we focus on high precision, mainly using Reid

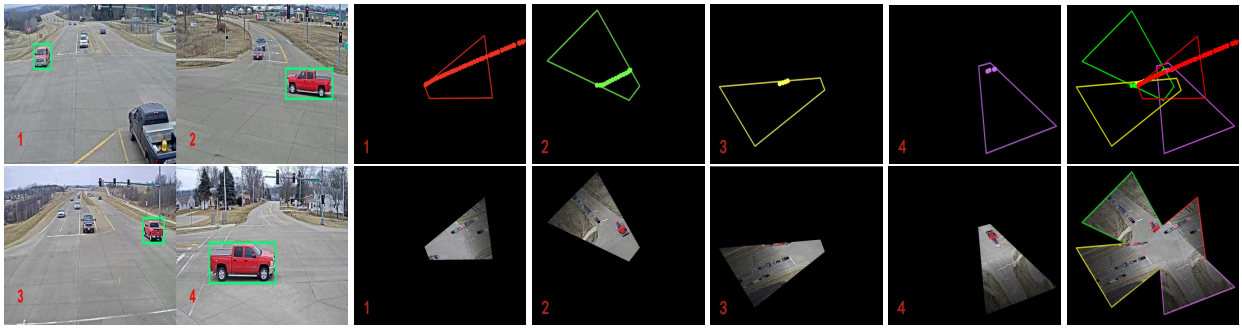

Figure 5. Single target recorded using multiple cameras; The figure demonstrated the matching of GPS-trajectory.

features to match, merging tracks with high similarity of Reid features, two matching tracks are required from different cameras, and can also be matched successfully when $IOU_{time_+} > 0$.

Stage 2 for crossroads scene For crossroads scene, a same car should be imaged by different cameras at the same time, and the direction of movement should be consistent. In addition, we find that for the same car in multi cameras, the images with the same vehicle orientation has higher similarity of deep features, while for the images with different orientation, we loosen the requirements of matching to improve the recall rate.

stage 2 for arterial road scene In arterial road scene, we also require consistency in the moving direction. But in this scene, the car orientation is less important. Because of the more various camera viewpoints, which leads to a relatively comprehensive features in orientation. So we reduce the weight of ReID feature importance, tracks that are tightly connected in both time and trajectory should be more likely same vehicle, in this case, we mainly focus on time and trajectory feature.

stage 3 post-processing for remaining single track For the final single track, matching requirements will be relaxed to improve recall. This part of the matching is also optimized for two different scenarios, and the method is the same as the second stage. The difference is that at this stage, only those single tracks are processed, and the matching threshold is improved.

3.5.3 Improve Localization Accuracy Based on Bird's-eye View

To match more tracks, we used bird's-eye view to analyze hard examples as shown in Figure 7. In previous section, we don't use overlapped information from different cameras.

Figure 6. We use 3D grids ROI(left) rather than road ROI(right) because of more accurate GPS location as described in official readme file.

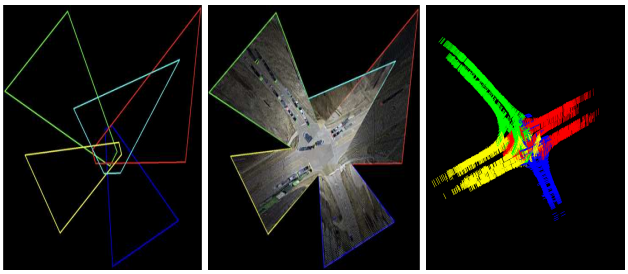

Figure 7. We visualize the 3D grid ROI in left and render real traffic scenarios in ROI in middle image, then we plot all vehicle trajectory of crossroads scene in right.

But this kind of information is very important for crossroad scene. In crossroad scene, the same car's GPS-trajectories viewed by different cameras, should be highly similar. In order to get more accurate GPS-trajectory, We have done three tricks on the GPS trajectory: 1. we choose the bottom center of the bounding box according to principle of projection. 2. As is shown in Figure 6, we only use the GPS coordinates in the given 3D grids ROI. 3. As is shown in Figure 8, we only use the trajectory information in the overlap of multi-camera. With high accurate GPS-trajectories, we can match the same tracks more easily as shown in Figure 5.

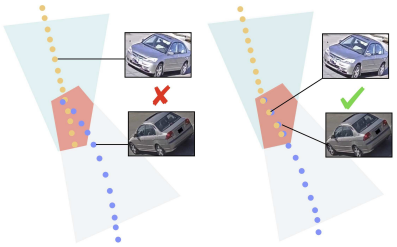

Figure 8. We can match two tracks with different appearance feature by using trajectory in the overlapping area.

4. Experimental Results

Our proposed methods are submitted for evaluation on the NVIDIA AI City Challenge 2019, in which we participate in track1: City-Scale Multi-Camera Vehicle Tracking. Our team achieves top performance in this track. The visualization of our qualitative performance is shown soon on our website. Detailed analyses on our performance are as follows. Experiments of Table 1 to Table 5 is based on the validation dataset including 184 vehicle identities, while the result in Table 6 is based on the test dataset including 482 vehicle identities.

4.1. Single-Camera Tracking

Method	IDF1	IDP	IDR
DS+YOLO	78.9%	–	–
DS+SSD	79.5%	–	–
DS+FRCNN	78.9%	–	–
TC+YOLO	79.1%	–	–
TC+SSD	79.7%	–	–
TC+FRCNN	78.7%	–	–
MO+YOLO	77.8%	–	–
MO+SSD	72.8%	–	–
MO+FRCNN	75.6%	–	–
ReID+GPS	82.1%	79.6%	84.8%
ReID+GPS+Post	**83.2%**	82.4%	84.0%

Table 1. Comparison among various methods with our MOT approach on the AI city Dataset.

The dataset contains 3.25 hours(195.03 minutes) of videos collected from 40 cameras spanning 10 intersections in a mid-sized U.S. city. The resolution of each video is at least 960p and the majority of the videos have a frame rate of 10 FPS. The task of each team is to detect and track targets with multiple cameras. For MTMC tracking, this track is evaluated based on IDF1 score[17] and rand the performance. IDF1 measures the ratio of correctly identified detections over the average number of ground-truth and computed detections. As shown in Table 1, compared with state-of-the-art Single-camera tracking algorithm, our method performs better.

4.2. Track-based ReID

We evaluate the effects of different strategies on Track-based ReID, and finally select the minimum feature distance between two tracks as the measure for Track-based ReID.

Strategy	F1	Precision	Recall
Average feature distance	46.3%	57.5%	38.8%
Mean of all feature distance	45.5%	55.6%	38.5%
Min of all feature distance	**54.1%**	**67.1%**	**45.3%**

Table 2. Comparison among various features for track-based ReID task.

4.3. GPS-trajectory direction optimization

We experiment on the MTMC task with simple head-tail directions and slice directions respectively.Our slice method has improved by 1.6%, and trajectory smoothness is also helpful to the improvement of results.

Method	IDF1	IDP	IDR
Head-tail Direction	68.6%	76.53%	62.21%
Slice Direction	70.2%	77.83%	63.97%
Smooth Slice Direction	**70.71%**	**78.42%**	**64.38%**

Table 3. Comparison among various direction features on the AI city Dataset.

4.4. Track Matching

In this place, we compare two matching strategies. Hierarchical matching consists of three different stages, while the simple matching only uses the second stage of hierarchical matching.

Method	IDF1	IDP	IDR
Simple Matching	63.72%	78.14%	53.79%
Hierarchical Matching	**70.71%**	**78.42%**	**64.38%**

Table 4. Comparison among various matching pipelines on the AI city Dataset.

4.5. Ablation Experiments

Ablation experiment result is shown in Table 5. In baseline, two tracks can be matched only when their $IOU_{time_+} \geq 0$, but we use different thresholds for $IOU_{time_+} \geq 0$ and $IOU_{time_+} < 0$ respectively, which is called soft time condition. Soft time condition improves IDF1 by 1.39%. In addition, we add vehicle orientation as

	Baseline					
Soft Time Condition		√	√	√	√	√
Vehicle Orientation			√	√	√	√
Single Track Matching				√	√	√
Parameter Searching					√	√
Accurate Localization						√
IDF1	65.53%	66.92%	67.78%	68.52%	70.71%	71.71%

Table 5. Ablation experiment result on AI City Challenge.

one of the matching features in crossroads scene, and the result is further improved. The vehicle orientation is obtained from movement direction in image. Then, parameter searching is used to search the best super parameter automatically for Algorithm 2. Single track matching and accurate localization have been described in 3.5.2 and 3.5.3. Finally, we got 71.71% result in this dataset as is shown in Table 5. There are 22 submissions in total for this track, the quantitive comparison of IDF1 scores across top ten teams is presented in Table 6.

Rank	Team ID	IDF1
1	21	70.59%
2	**Ours**	**68.65%**
3	12	66.53%
4	53	66.44%
5	97	65.19%
6	59	59.87%
7	36	49.24%
8	107	45.04%
9	104	33.69%
10	52	28.50%

Table 6. Comparison of performance with other teams.

5. Conclusion

In this paper, we propose a spatio-temporal consistency and hierarchical matching method for vehicle tracking in across cameras. Our intuition is to combine the spatial and time feature for characterizing the targets and leverage a bottom-up hierarchical matching strategy to compare targets in different cameras. Our framework includes several components for city-scale multi-camera vehicle tracking. Firstly, we leverage FPN to detect vehicles and object tracking methods to track vehicles in single camera. Furthermore, a spatio-temporal consistency method is proposed to extract features of target vehicle. Finally, a bottom-up hierarchical match strategy is proposed to match targets in different cameras. Our method is validated in City-Scale Multi-Camera Vehicle Tracking task at the NVIDIA AI City Challenge 2019, and the experiment results demonstrate advantages of our proposed method.

References

[1] Christos-Nikolaos E Anagnostopoulos, Ioannis E Anagnostopoulos, Ioannis D Psoroulas, Vassili Loumos, and Eleftherios Kayafas. License plate recognition from still images and video sequences: A survey. *IEEE Transactions on intelligent transportation systems*, 9(3):377–391, 2008. 2

[2] Heng Fan, Liting Lin, Fan Yang, Peng Chu, Ge Deng, Sijia Yu, Hexin Bai, Yong Xu, Chunyuan Liao, and Haibin Ling. Lasot: A high-quality benchmark for large-scale single object tracking. *arXiv preprint arXiv:1809.07845*, 2018. 2

[3] Thomas Fortmann, Yaakov Bar-Shalom, and Molly Scheffe. Sonar tracking of multiple targets using joint probabilistic data association. *IEEE journal of Oceanic Engineering*, 8(3):173–184, 1983. 2

[4] Ross Girshick. Fast r-cnn. In *Proceedings of the IEEE international conference on computer vision*, pages 1440–1448, 2015. 2

[5] Ross Girshick, Jeff Donahue, Trevor Darrell, and Jitendra Malik. Rich feature hierarchies for accurate object detection and semantic segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 580–587, 2014. 2

[6] Priya Goyal, Piotr Dollár, Ross Girshick, Pieter Noordhuis, Lukasz Wesolowski, Aapo Kyrola, Andrew Tulloch, Yangqing Jia, and Kaiming He. Accurate, large minibatch sgd: Training imagenet in 1 hour. *arXiv preprint arXiv:1706.02677*, 2017. 3

[7] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016. 3

[8] Alexander Hermans, Lucas Beyer, and Bastian Leibe. In defense of the triplet loss for person re-identification. *arXiv preprint arXiv:1703.07737*, 2017. 3

[9] Harold W Kuhn. The hungarian method for the assignment problem. *Naval research logistics quarterly*, 2(1-2):83–97, 1955. 3

[10] Tsung-Yi Lin, Piotr Dollár, Ross Girshick, Kaiming He, Bharath Hariharan, and Serge Belongie. Feature pyramid networks for object detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2117–2125, 2017. 3

[11] 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*, pages 21–37. Springer, 2016. 2

[12] Wenqian Liu, Octavia Camps, and Mario Sznaier. Multi-camera multi-object tracking. *arXiv preprint arXiv:1709.07065*, 2017. 2

[13] Xinchen Liu, Wu Liu, Huadong Ma, and Huiyuan Fu. Large-scale vehicle re-identification in urban surveillance videos. In *2016 IEEE International Conference on Multimedia and Expo (ICME)*, pages 1–6. IEEE, 2016. 2

[14] Joseph Redmon, Santosh Divvala, Ross Girshick, and Ali Farhadi. You only look once: Unified, real-time object detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 779–788, 2016. 2

[15] Joseph Redmon and Ali Farhadi. Yolo9000: better, faster, stronger. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 7263–7271, 2017. 2

[16] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. In *Advances in neural information processing systems*, pages 91–99, 2015. 2

[17] Zheng Tang, Milind Naphade, Ming-Yu Liu, Xiaodong Yang, Stan Birchfield, Shuo Wang, Ratnesh Kumar, David Anastasiu, and Jenq-Neng Hwang. Cityflow: A city-scale benchmark for multi-target multi-camera vehicle tracking and re-identification. *arXiv preprint arXiv:1903.09254*, 2019. 2, 7

[18] 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 *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, pages 108–115, 2018. 2

[19] Jasper RR Uijlings, Koen EA Van De Sande, Theo Gevers, and Arnold WM Smeulders. Selective search for object recognition. *International journal of computer vision*, 104(2):154–171, 2013. 2

[20] Peng Wang and Qiang Ji. Robust face tracking via collaboration of generic and specific models. *IEEE transactions on image processing*, 17(7):1189–1199, 2008. 2

[21] Rey Reza Wiyatno and Anqi Xu. Physical adversarial textures that fool visual object tracking. *arXiv preprint arXiv:1904.11042*, 2019. 2

[22] Fu Xiong, Yang Xiao, Zhiguo Cao, Kaicheng Gong, Zhiwen Fang, and Joey Tianyi Zhou. Towards good practices on building effective cnn baseline model for person re-identification. *arXiv preprint arXiv:1807.11042*, 2018. 3

[23] Rui Yao, Guosheng Lin, Shixiong Xia, Jiaqi Zhao, and Yong Zhou. Video object segmentation and tracking: A survey. *arXiv preprint arXiv:1904.09172*, 2019. 2