Multi-camera Vehicle Tracking and Re-identification on AI City Challenge 2019

Yucheng Chen[1], Longlong Jing[2], Elahe Vahdani[2], Ling Zhang[3], Mingyi He[1], and Yingli Tian[2,3*]
[1]Northwestern Polytechnical University, Xi'an, China, 710129
[2]The Graduate Center, The City University of New York, NY, 10016
[3]The City College, The City University of New York, NY 10031

Abstract

In this work, we present our solutions to the image-based vehicle re-identification (ReID) track and multi-camera vehicle tracking (MVT) tracks on AI City Challenge 2019 (AIC2019). For the ReID track, we propose an enhanced multi-granularity network with multiple branches to extract visual features for vehicles with different levels of grains. With the help of these multi-grained features, the proposed framework outperforms the current state-of-the-art vehicle ReID method by 16.3% *on Veri dataset. For the MVT track, we first generate tracklets by Kernighan-Lin graph partitioning algorithm with feature and motion correlation, then combine tracklets to trajectories by proposed progressive connection strategy, finally match trajectories under different camera views based on the annotated road boundaries. Our MVT and ReID algorithms are ranked the* 10 *and* 23 *in MVT and ReID tracks respectively at the NVIDIA AI City Challenge 2019.*

1. Introduction

With the advances of Intelligent Transportation System, the city-scale transportation data analysis including multiple vehicle tracking (MVT) across cameras with non-overlapping regions and vehicle re-identification have drawn more and more attention in the recent years [18].

The vehicle ReID task aims at identifying moving vehicles by matching a given car across different non-overlapping cameras. The appearance of the vehicle may significantly change due to lighting variation, occlusion, viewing angle, or scale which makes ReID a challenging task. Another challenge of vehicle ReID is to label accurate ground-truth for vehicle ReID datasets because of the limited quality and viewing angles of traffic videos and the occluded or vague license plates. Moreover, the trained models on existing vehicle datasets may not work well for the new testing environment because the cityscape differs from city to city and car models change from time to time [24]. Hence, the features learned from one dataset may not be directly applicable to another dataset.

Comparing vehicle ReID with person ReID, the former is considered more challenging because of small inter-class variability and large intra-class variability of vehicles [18]. In other words, the number of car models is small which indicates many cars may have same color and in same model. It is difficult to distinguish cars with the same model and color without other detailed information such as license plate. In contrast, people are easier to distinguish because of having more distinct features including face and clothing. In addition, vehicles move along a fixed direction without rotation and the key information of the vehicle might be invisible in some frames. However, for humans, the key features like face show up from time to time [6].

The objective of MVT is to discover the trajectories of all vehicles through the videos under multiple camera views. City-scale tracking of vehicles from multiple cameras is challenging due to the following reasons. First, many cars are in the same model and the similar appearance which makes them harder to distinguish. Second, occlusion happens frequently in busy traffic flow which can lead to a high number of identity switches between different cars. Moreover, the viewpoints or appearance of the same car can vary largely in different cameras and under different lighting conditions. To mitigate the above challenges, some methods utilize the 3D information of the vehicles or depth data for more reliable predictions. The 3D information can be retrieved using the back-projection of vehicle positions from the camera projection matrix [17]. The camera parameters can also be optimized using evolutionary algorithms [19].

In general, MVT methods can be categorized into two groups: 1) global optimization methods and 2) online methods. In the global optimization methods, the observations of vehicles are grouped into tracklets based on spatial-temporal continuity and the cost of data association (vehicles to track identities) is minimized across the entire video [6]. In contrast, in the online methods, the trajectory of each target is constructed frame-by-frame in order to find the ap-

*corresponding author. Email: ytian@ccny.cuny.edu

propriate matching model to connect the detection results of the current frame to the track identities from the previous frames.

In this paper, we describe the frameworks developed for vehicles ReID and MVT tasks respectively. Our ReID network includes several branches to extract feature with various levels of grains. In our MVT framework, we first consider tracklets generation as a graph partitioning problem solved by KernighanLin algorithm with feature and motion correlation, then propose a progressive connection strategy to generate trajectories from tracklets, finally match these trajectories under different camera views based on manually annotated road boundaries.

2. Related work

2.1. Re-identification

Person ReID training datasets are often weakly diverse and the infrequent detailed information could be ignored in the global feature learning. Hence, considering different scales of the input images and capturing multiple granularities have drawn more attention in the recent state-of-the-art methods such as Deep Pyramid Feature Learning (DPFL) [5] and Multiple Granularities Network (MGN) [20]. Chen *et al.* proposed to learn the most discriminative visual features of different image scales. The network of DPFL includes multiple branches to model input image scaled to different resolutions and then a fusion branch is used to learn the optimal integration of scale-specific features for the complementary information across different scales. Wang *et al.* proposed to aggregate salient features from the global and local parts of the body [20]. The network consists of a global branch to capture the global features and two local branches to obtain the local feature representations with multiple granularities. Softmax and triplet loss are deployed for classification and metric learning, respectively.

A list of good practices is proposed in [1] to design and train an efficient image representation model for person ReID. The key practices include pre-training for identity classification, sufficiently large image resolution, state-of-the-art base architecture, hard triplet mining, and dataset augmentation with difficult examples. These techniques such as triplet mining and ID classification have been employed in vehicle ReID [2, 11, 10]. Bai *et al.* [2] proposed an online grouping method to partition the samples of each vehicle ID into a set of groups (samples with similar attributes in the same cluster) and incorporate the intra-class variance with the triplet loss. The triplet loss is replaced with Coupled Clusters Loss (CCL) in [11] to minimize the distances of the same vehicle images and maximize those of other vehicles. An extensive evaluation of contrastive and triplet loss for vehicle ReID task is provided in [10].

As mentioned before, one inherent challenge in the vehicle identification and tracking tasks is the variety of appearance from different viewing angles. To handle this issue, Zhou *et al.* proposed a viewpoint invariant framework by deploying a viewpoint-aware attention model and the adversarial training architecture [27]. For the images captured from arbitrary viewpoints, the single-view features are transferred to a global multi-view feature representation. Wang *et al.* proposed a framework with orientation invariant feature embedding and spatial-temporal regularization [21]. Yan *et al.* proposed a multi-grain based list ranking (MGLR) approach [25]. They maintain a list of multi-grain images and rank them based on the multi-grain relationship, considering the possibility of any permutations after ranking. The results in CityFlow showed the effectiveness of the combination of hard triplet loss [7] with cross-entropy loss [16] and DenseNet121 [8] to achieve the best performance on CityFlow-ReID sub-dataset.

2.2. Multi-Object Tracking

Multi-object tracking (MOT) is a relatively new task with only a few papers. Among the online methods for MOT, SORT [4] achieved the best rank in 2D MOT 2015 by deploying Kalman filter and Hungarian method for motion prediction and data association, but ignores the appearance features beyond the detection component and does not handle occlusion well. DeepSORT [23] aggregates the appearance features (extracted from a CNN) along with motion information from Kalman filter and also used the Hungarian algorithm for data association. This method reduced the identity switches by 45% compared with SORT [4]. MOANA [17] employed an adaptive appearance model to encode the long-term appearance change along each trajectory in an online manner. With utilizing the long-term history of appearance, this model is more robust against the change in lighting condition and object pose, resulting in a better performance when occlusion happens. The winner of AI City Challenge 2018 [19] proposed an offline framework to address single camera tracking and vehicle ReID. Their framework encodes the long-term appearance change of each target using a histogram-based adaptive model. The tracking is done by clustering the tracklets in a bottom-up approach using several semantic features such as trajectory smoothness, velocity change, and temporal information. They incorporate the features extracted from a CNN pre-trained on CompCars benchmark for the re-identification.

3. Methodology

3.1. Vehicle Re-Identification

Framework: The architecture of the proposed network for vehicle ReID is shown in Fig. 1. The backbone of the network is a convolution neural network that pre-trained on ImageNet dataset, and the five parallel independent

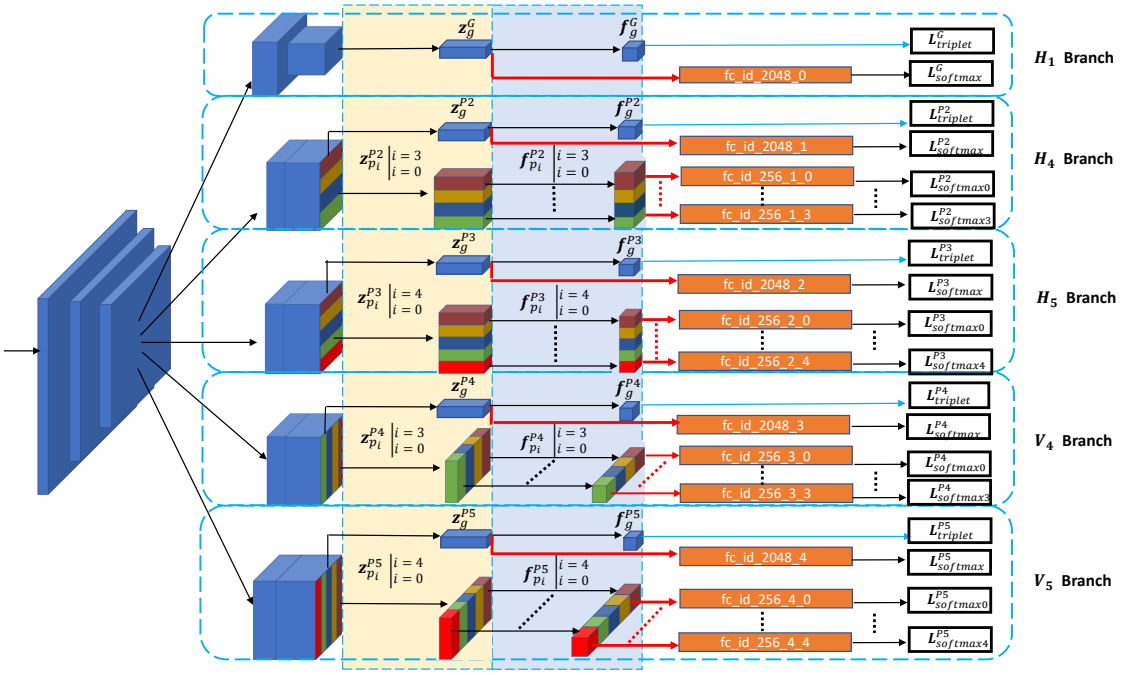

Figure 1. The architecture of the proposed network for vehicle ReID task. The backbone network is split into five independent branches after the 4_{th} convolution layer. During vehicle ReID testing, only the global features of the five branches are used as the representation for each vehicle image.

branches extract features with different levels.

The settings of the five branches are shown in Table 1. The branch $H1$ consists of a stride-2 convolution layer followed by a global max pooling and 1×1 convolution layer to reduce the feature dimension from 2,048 to 256. With the global max pooling over the whole feature map, this branch is able to capture the global features. The branches $H4$, $H5$, $V4$, and $V5$ consist of a non-stride convolution layer followed by a stride max pooling to evenly dive the feature map horizontally or vertically to stripe and the fine-grained features are then extracted from each stripe. The extracted fine grained features are then fed to fully connected layer to reduce the dimension to 256.

During training, both the global and the fine grained features are employed for ID classification with cross entropy loss and metric learning with triplet loss. During testing, only the global features from the five branches are concatenated for ReID.

Loss Function: For the task of person [1, 20] and vehicle re-identification [18], both the cross entropy loss and the triplet loss have been widely used for optimizing the networks. Following others, the softmax loss is chosen for the classification and the triplet loss is chosen as the loss for metric learning during the training phase.

For discriminative learning, the problem has been formulated as a multi-class classification while the label for each image is the vehicle ID. For each feature $\mathbf{f}_i$ that used for discriminative learning, the softmax loss is formulated as:

Branch	Part No.	Max Size	Dims	Feature
$H1$	1	8×8	256	$\mathbf{f}_g^G$
$H4$	4	16×5	256*3+256	$\{\mathbf{f}_{p_i}^{P2}\|_{i=0}^{1}\}, \mathbf{f}_g^{P2}$
$H5$	5	16×4	256*4+256	$\{\mathbf{f}_{p_i}^{P3}\|_{i=0}^{2}\}, \mathbf{f}_g^{P3}$
$V4$	4	5×16	256*3+256	$\{\mathbf{f}_{p_i}^{P4}\|_{i=0}^{1}\}, \mathbf{f}_g^{P4}$
$V5$	5	4×16	256*4+256	$\{\mathbf{f}_{p_i}^{P5}\|_{i=0}^{2}\}, \mathbf{f}_g^{P5}$

Table 1. Comparison of the settings for the five branches in the proposed network. "Branch" refers to the name of branches. "Part No." refers to the number of partitions on feature maps. "Max Size" refers to the size of max pooling for each branch. "Dims" refers to the dimension and number of features for the output representations. "Feature" means the symbols for the output feature representations of each branch.

$$L_{softmax} = -\sum_{i=1}^{N} \log \frac{e^{\mathbf{W}_{y_i}^T \mathbf{f_i}}}{\sum_{k=1}^{C} e^{\mathbf{W}_k^T \mathbf{f_i}}}, \quad (1)$$

where $\mathbf{W}_k$ corresponds to a weight vector for class k, with the size of mini-batch in training process N and the number of classes in the training dataset C.

Among all the learned features, the softmax loss is computed over the global features of all the five branches before 1×1 convolution reduction $\{\mathbf{z}_g^G, \mathbf{z}_g^{P2}, \mathbf{z}_g^{P3}, \mathbf{z}_g^{P4}, \mathbf{z}_g^{P5}\}$ and all the fine-grained features extracted from all the stripes of the five branches. The softmax over global features and the

fine-grained features forces network to focus on the most discriminative parts of vehicles.

In addition to the softmax loss, all the global features are trained with triplet loss to enhance ranking performance. Following others [20], the batch-hard triplet loss is employed which is an improved version of the semi-hard triplet loss. This loss function is formulated as follows:

$$L_{triplet} = -\sum_{i=1}^{P}\sum_{a=1}^{K}[\alpha + \max_{p=1...K} \|\mathbf{f}_a^{(i)} - \mathbf{f}_p^{(i)}\|_2 - \min_{\substack{n=1...K \\ j=1...P \\ j\neq i}} \|\mathbf{f}_a^{(i)} - \mathbf{f}_n^{(j)}\|_2]_+, \quad (2)$$

where $\mathbf{f}_a^{(i)}, \mathbf{f}_p^{(i)}, \mathbf{f}_n^{(i)}$ are the features extracted from anchor, positive, and negative samples receptively, and α is the margin hyper-parameter to control the differences of intra and inter distances. In the vehicle ReID, the positive and negative pairs refer to vehicles with same or different identity for the anchor vehicle. With the loss function, only the hardest positive and negative pairs in a mini-batch with P selected identities and K images from each identity are selected to obtain the loss for the batch.

Evaluation Metrics: Following other work [12, 18], the mean-average-precision (*mAP*) and *top-k* are employed for performance evaluation and comparison with other approaches. During the evaluation phase, a set of query images and a set of gallery images are given, and the goal is to find the top-k nearest images from the gallery images which match the query image. For each query image q, the average precision is defined as:

$$AP(q) = \frac{\sum_k P(k) \times \delta_k}{N_{gt}(q)}, \quad (3)$$

where $P(k)$ represents the precision at rank k, $N_{gt}(q)$ is the total number of true retrievals for q. δ_k is 1 when the matching of query image q is correct and the rank $<= k$. mAP is then computed as average over all query images.

3.2. Multi-camera Vehicle Tracking

3.2.1 Framework

The pipeline of our proposed framework for multi-camera multi-target tracking is presented in figure 2. It consists of four components which are described in the following subsections. (a) Given video streams, the Mask-RCNN network is employed to extract bounding boxes and masks of vehicles from videos. (b) After refinement for reducing false positive candidates, the proposed ReID network is trained to extract feature vector for each detected candidate. Meanwhile, camera calibration is applied to obtain the GPS location from 2D-GPS projection for each detected vehicle candidate. The appearance features and location information of detected vehicle candidates are converted into correlations for tracklet generation. (c) A progressive connection method is proposed to incorporate tracklets to a whole trajectory and revise two kinds of matching deviations. (d) Finally, the identity matching across different cameras is constrained with our manually annotated road boundaries and determined by feature distance.

3.2.2 Vehicle Detection and Refinement

The vehicle detection results provided by AI City challenge (generated by Mask-RCNN in the MOTChallenge) including both bounding boxes and masks are employed. However, the initial detection results are very noisy with tiny, parked and occluded vehicles. We further refine the detection results by reducing the false positives with small impact on recall. As shown in Table 2 on all cameras in training data, the refined vehicle detection results are significantly improved compared to the original results.

	F1	Recall	Precision
w/o refinement	0.253	0.994	0.156
w/ refinement	0.695	0.993	0.560

Table 2. Detection results, without and with refinement.

Basic constraints: We add basic constraints for bounding boxes, including height, width, area, aspect ratio of bounding box, ratio of effective area in the bounding box (both mask and foreground detection by ViBe [3] are used), and distance of the center to edges, etc. Above parameters are chosen over all training set. Also, we use NMS with confident score threshold of 0.5 to remove overlapped bounding boxes.

Background Detection. To generate the background without moving vehicles, for each pixel p, we select all the frames in which the pixel does not belong to any bounding box. Then, the background value of p, is set as the average value of the pixels in the selected frames. If p is inside a bounding box for all frames, it means that p is part of a car which belongs to the background. Second, for each detected region (bounding box), we crop the bounding box and the corresponding region in the background image to calculate the Structural Similarity Index (SSIM), as well as the SSIM by cropping the mask from the bounding box and the corresponding mask in the background. The thresholds for SSIM based on bounding box and mask are 0.6 and 0.65, respectively.

3.2.3 Tracklet generation

Same as [15], tracklet generation is treated as a graph partitioning problem. Since most vehicles move fast, a 4-frame

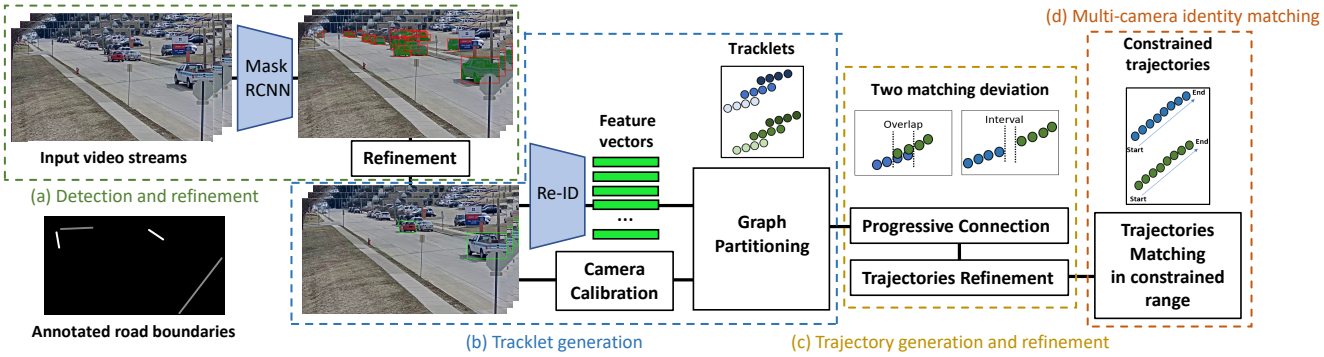

Figure 2. The pipeline of our proposed framework for multi-camera vehicle tracking (MVT) task. Annotated road boundaries are employed as a constraint for trajectories.

sliding with a stride of 2 is employed to prevent the detected bounding box centers to interleave together. In each 4-frame sliding window, the detected results are treated as a weighted graph $G = (V, E, W)$, where V is a node set of bounding boxes (indicates the centers or footpoints of bounding boxes), E is an edge set linking nodes and W is a correlation matrix. The KernighanLin algorithm [9] is employed to handle graph partitioning with W composed by feature correlation W_a, pixel-level motion correlation W_{pm}, and GPS-level motion correlation W_{gm}. Before that, Hierarchical Clustering [22] is employed to group detected results into small groups to narrow the range of graph partitioning and reduce the amount of calculation based on GPS Euclidean distance between vehicles.

Feature correlation. The feature used for tracking is extracted by our network trained on the Veri dataset [12] or CityFlow dataset [18]. Both global and local features from the fully connected layers are extracted and concatenated into a 3,328-dimensional feature vector for each vehicle. Each w_{ij} in the feature correlation matrix W_f represents the correlation between two feature vectors v_i and v_j, defined as $w_{ij} = \frac{t_a - \|v_i, v_j\|_2}{t_a}$, where the threshold t_a is the mean value of mean positive pairs distance and mean negative pairs distance of all training data. Strongly correlated feature vector pairs receive w_{ij} near 1.

Pixel-level motion correlation. Since the 4-frame processing window is very short for fast moving vehicles, we assume the tracklet as a straight line. The velocity of each bounding box center is estimated by nearest center points. A linear motion model is employed to do forward-backward prediction of each center. The correlation between two centers is defined as $w_{pm} = \alpha(t_{pm} - e_f - e_b)$, where e_f and e_b represent forward and backward prediction errors respectively, t_m is obtained from training data to separate positive and negative evidence, α is a scaling factor. The range of w_{pm} is $-\infty$ to $\alpha \times t_m$.

GPS-level motion correlation. Similar as pixel-level motion correlation, we further employ W_{gm} to represents the motion correlation of detected results in GPS space, which balances the influence of image perspective principle to estimate motion. The middle point of the bottom edge of each bounding box is chosen as the footpoint of corresponding object to calculate GPS position.

3.2.4 Trajectory generation and refinement

We propose a progressive connection method to incorporate tracklets to a whole trajectory. Since the generated 4-frame tracklets have 2-frame overlap with previous and next tracklets, a simple matching score is employed to measure how many bounding box centers are perfectly matched between two tracklets. Specific algorithm steps are described as *Algorithm 1.*

When only using progressive connection for all tracklets, as shown in Figure 3, two kinds of matching deviations may happen. False positive detections of tracklets may bring trajectories to a wrong direction. False negative detections and occlusion may occur interval which break a trajectory into two trajectories. For the case (a), the corresponding center distance is averaged and normalized by mean diagonal length of bounding boxes, a threshold 0.15 is set to distinguish whether these two trajectories should be fuse together or not. For the case (b), Gaussian regression is applied for both trajectories to predict center location in the interval, center distance is computed same as case (a), a threshold 0.3 is used to decide whether these two trajectories should share same identity or not.

To further improve the results of trajectory generation, the short trajectories and the trajectories whose center points are almost not moving are deleted.

3.2.5 Multi-camera identity matching

To compare the trajectories under different camera views, all features of one trajectory are averaged as the representa-

Algorithm 1 Progressive Connection

tl: tracklet; tj: trajectory; $score$: matching score; f_{dist}: feature distance between two tracklets;
for each 4-frame window sliding with 2 frames **do**
  Find all tl within the sliding window
  if no active tj **then**
    Convert tl to new tj, label new identity
  else
    for each active tj **do**
      Compute $score$ to all not attached tl
      if one tl with $score$ of 2 **then**
        Find tl with $score$ of 2 and attach to tj
      else if one tl with $score$ of 1 **then**
        Find tl with $score$ of 1 and attach to tj
      else if two tls with $score$ of 1 **then**
        Choose tl with lower f_{dist} and attach to tj
      else if all tls with $score$ of 0 **then**
        End tj
      end if
    end for
    if exist tl with no tj to attach **then**
      Convert tl to new tj, label new identity
    end if
  end if
end for

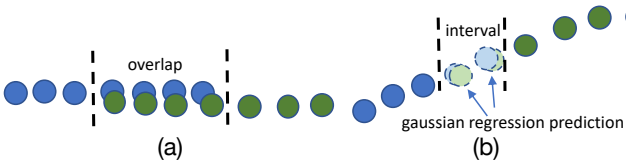

Figure 3. Two kinds of matching deviations: (a) Overlapped trajectories and (b) interval trajectories.

tion. Since arbitrary inter-camera identity matching is very inefficient and impractical, here, three constraints are applied to narrow the matching range which are direction constraint, camera location constraint and road boundary constraint. In the matching range, two trajectories with the lowest feature distance will share the same identity. After applying these three constraints, the trajectories obtain the information of camera locations and the estimated time of the vehicle to appear in the next camera view.

Direction constraint. The direction of each trajectory is determined by the order in which the bounding box centers appear.

Camera location constraint. Based on camera locations, the approximate time running between different cameras can be estimated,

Road boundary constraint. For each camera view, the road boundaries are manually annotated as shown in Figure 3. The value of boundary represents the next show up camera view through the current direction.

Figure 4. Examples of annotated road boundaries in background.

3.2.6 Evaluation Metrics

In [14], the authors modeled the ground-truth and computed trajectories with a biparitie graph $G = (V_T, V_C, E)$ where the vertex set V_T has one regular node for each true trajectory and one false positive node for each computed trajectory. In other hand, vertex set V_C has one regular node for each computed trajectory and one false negative node for each true trajectory. A bipartite match associates one ground-truth trajectory to exactly one computed trajectory by minimizing the number of mismatched frames over the true and computed data.

$IDTP$ (True Positive ID) includes the pairs of computed identity and ground-truth identity which are matched. Similarly, $IDFP$ and $IDFN$ stand for false positive ID and false negative ID, respectively. IDP, IDR and IDF_1 is defined as following measurement similar to detection evaluation metrics.

$$IDP = \frac{IDTP}{IDTP + IDFP} \tag{4}$$

$$IDR = \frac{IDTP}{IDTP + IDFN} \tag{5}$$

$$IDF_1 = \frac{2\,IDTP}{2\,IDTP + IDFP + IDFN} \tag{6}$$

4. Experiments

4.1. Dataset

Veri776: The Veri776 dataset is proposed by [12] and is a widely used benchmark dataset for vehicle re-identification. The dataset consists of $40,000$ bounding box annotations of 776 cars (identities) across 20 cameras in traffic scenes. Each vehicle is captured in $2-18$ cameras in various viewpoints and varying illuminations. Our model is evaluated on this dataset to obtain a baseline for the vehicle ReID task of AIC2019.

Granularity	Backbone	Resolution	MAP	Rank-1	Rank-5	MAP (RK)	Rank-1 (RK)	Rank-5 (RK)
$H1$	ResNet50	224	**66.7**	91.8	96.1	**71.4**	92.9	94.9
$H1, H2$	ResNet50	224	**74.4**	94.9	97.3	**76.9**	95.4	96.2
$H1, H2, H3$	ResNet50	224	**77.7**	95.3	97.1	**79.8**	95.5	96.5
$H1, H2, H3, H4$	ResNet50	224	**79.5**	95.5	97.6	**81.5**	96.1	97.4
$H1, H2, H3, H4$	ResNet50	256	**80.5**	96.1	98.0	**82.4**	96.4	97.4
$H1, H2, H3, H4$	SEResNext50	256	**81.9**	96.2	97.9	**83.9**	96.9	97.7

Table 3. Impacts of different branches to the performance. The mean average precision (mAP) on Veri776 is reported. "RK" refers to the results improved by performing Re-Ranking.

AI City 2019 Dataset: CityFlow [18] is a new dataset for AI City Challenge 2019 and is considered as the largest scale dataset in terms of spatial coverage and number of cameras. The dataset is collected from 40 cameras spanning across 10 intersections with a duration of 3.25 hours. It includes $229,680$ bounding boxes of 666 vehicle identities where each passes through at least 2 cameras.

4.2. Implementation Details

During training for ReID, each image is resized into 320×320 and then a patch with a size of 256×256 is randomly selected from each image as a training image. Each image is also applied with three types of data augmentation: (1) horizontal flip with 50% probability, (2) rotate with a degree randomly sampled from -10 to 10 degree, (3) set the values of pixels within a randomly selected patch with 0. The training is optimized by AMSGrad [13] using 120 epochs and with a batch size of 120. The initial learning rate is set to 0.0003 and is decayed by 0.1 at 20 and 40 epochs.

4.3. Results of Image-based Re-Identification

Impact of granularity: To verify the effectiveness of each branch in the proposed network, ablation experiments with different component settings are conducted on the Veri dataset. As shown in the Table 3, the performance with the only global features is 66.7% on Veri datasets. The finer branch can significantly boost performance. For example, the branch $H2$ can boost the performance of $H1$ branch by 7.7%, while $H3$ can further boost the performance by 3.3%. Also, the re-ranking [26] can significantly improve the performance for all the networks.

Compare with others on the Veri dataset: We compare the proposed methods with the current state-of-the-art methods for vehicle ReID on the Veri and AIC dataset and show the performance in Table. 4. Our model outperforms the current state-of-the-art Vehicle ReID method on Veri-776 dataset by 16.3% on mAP.

Visualization of the results: The qualitative results on CityFlow dataset are shown in Fig. 5. The top-10 ranking lists for six query images are visualized.

Method	mAP	Rank-1
OIFE [21]	48.0	89.4
VAMI [27]	50.1	77.0
GSTE [2]	59.5	96.2
MoV1BS [10]	67.6	90.2
Ours	**83.9 (+16.3%)**	**96.9 (+6.7%)**

Table 4. Quantitative evaluation of the state-of-the-art metric learning methods in vehicle ReID on Veri dataset. Our model outperforms the state-of-the-art method on Veri dataset by 16.3% on mAP.

4.4. Ablation Study of the Single-camera Tracking

To evaluate the effectiveness of different components in the proposed single-camera tracking method, ablation studies are conducted and the tracking performance for single camera are reported and compared. During all the experiments, the features are extracted by the proposed ReID network which is only trained on Veri776 dataset, and all the training set of CityFlow are used as the validation set. Trained with a cross-domain vehicle dataset, the results of single-camera vehicle tracking on the CityFlow training set are shown in Table 5.

Method	IDF1	IDP	IDR
$baseline$	0.594	0.449	0.878
$baseline + G_g$	0.605	0.459	0.890
$baseline + G_g + G_m$	0.630	0.477	0.926
$baseline + G_g + G_m + Tj_r$	0.657	0.499	0.962
$baseline + G_g + G_m + Tj_r + Tj_d$	0.755	0.647	0.907

Table 5. The impacts of different components in the proposed single-camera tracking method. The results are obtained by testing the algorithm on the training split of CityFlow dataset while the feature extractor is trained on Veri776 dataset. The $baseline$ only employs visual features and pixel-level motion correlation for tracklet generation without trajectory refinement, the G_g represents GPS-based Hierarchical grouping before tracklet generation, the G_m represents GPS-based motion correlation for tracklet generation, the Tj_r represents trajectory matching deviations refinement, the Tj_d represents short and motionless trajectory deletion.

The experiments are conducted to evaluate the components including: GPS-based grouping (G_g), GPS-based motion correlation (G_m), Trajectory refinement (Tj_r), and

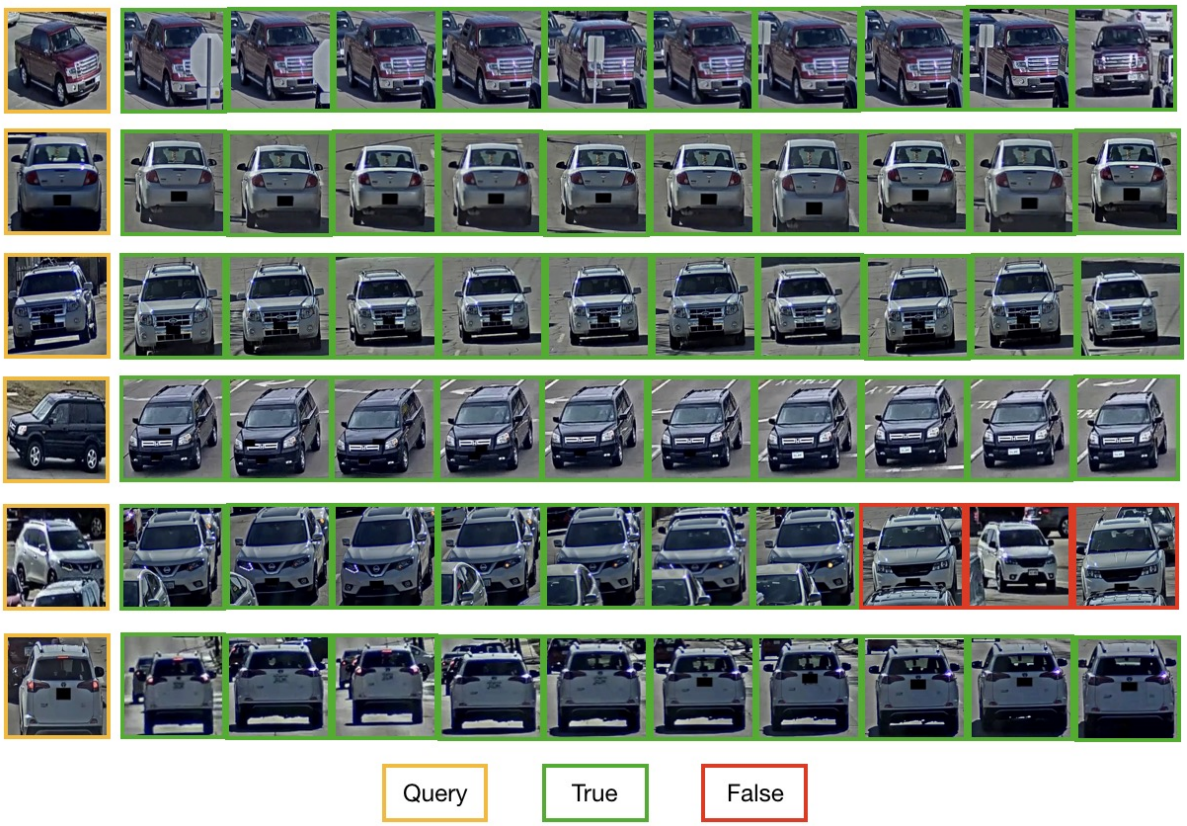

Figure 5. Top-10 ranking lists for six query images on CityFlow dataset by our network. The images with green boundaries belong to the same identity as the query, and images with red borders do not.

Trajectories (Tj_d). Most of the components can significantly boost the tracking performance on CityFlow dataset. Among all the components, the Tj_d can boost the performance by 0.093 on CityFlow dataset since it can effectively reduce the false positive of trajectories.

4.5. Results on AIC2019

Table 6 lists the ranks of our team and the results of our team as well as the top three and the last teams on MVT and ReID tasks of AIC2019. Our MVT and ReID algorithms are ranked the 10 out of 22 and the 23 out of 84 teams in MVT and ReID tracks respectively. Our ReID method is based on images and does not use the temporal information of the tracklets, therefore, the performance can be further improved by utilizing the temporal information of tracklets.

MVT			ReID		
Rank	Team ID	IDF	Rank	Team ID	mAP
1	21	0.7059	1	59	0.8554
2	49	0.6865	2	21	0.7917
3	12	0.6653	3	97	0.7589
10	**52**	**0.2850**	**23**	**52**	**0.4096**
22	45	0.0326	84	133	0.0003

Table 6. Results and ranks of our proposed methods on MVT and ReID tasks of AIC2019.

5. Conclusion

To handle vehicle re-identification (ReID) and multi-camera vehicle tracking (MVT) tracks on AI City Challenge 2019, we have proposed an enhanced multi-granularity network and designed an offline tracking framework. The proposed ReID network outperforms the current state-of-the-art vehicle ReID methods by 16.3% on the Veri dataset and ranks the 23rd place in AIC2019, while the framework for MVT tracking ranks the 10th place.

6. Acknowledgement

This material is based upon work supported by the National Science Foundation under award number IIS-1400802 and Natural Science Foundation of China (61420106007, 61671387). Yucheng Chen's contribution was made when he was a visiting student at the City University of New York, sponsored by the Chinese Scholarship Council.

References

[1] Jon Almazan, Bojana Gajic, Naila Murray, and Diane Larlus. Re-id done right: towards good practices for person re-identification. *arXiv preprint arXiv:1801.05339*, 2018.

[2] Yan Bai, Yihang Lou, Feng Gao, Shiqi Wang, Yuwei Wu, and Ling-Yu Duan. Group-sensitive triplet embedding for vehicle reidentification. *IEEE Transactions on Multimedia*, 20(9):2385–2399, 2018.

[3] Olivier Barnich and Marc Van Droogenbroeck. Vibe: A universal background subtraction algorithm for video sequences. *TIP*, 20(6):1709–1724, 2011.

[4] Alex Bewley, Zongyuan Ge, Lionel Ott, Fabio Ramos, and Ben Upcroft. Simple online and realtime tracking. In *ICIP*, pages 3464–3468. IEEE, 2016.

[5] Yanbei Chen, Xiatian Zhu, and Shaogang Gong. Person re-identification by deep learning multi-scale representations. In *ICCV*, pages 2590–2600, 2017.

[6] Weitao Feng, Deyi Ji, Yiru Wang, Shuorong Chang, Hansheng Ren, and Weihao Gan. Challenges on large scale surveillance video analysis. In *CVPRW*, pages 69–76, 2018.

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

[8] Gao Huang, Zhuang Liu, Laurens Van Der Maaten, and Kilian Q Weinberger. Densely connected convolutional networks. In *CVPR*, pages 4700–4708, 2017.

[9] Brian W Kernighan and Shen Lin. An efficient heuristic procedure for partitioning graphs. *Bell system technical journal*, 49(2):291–307, 1970.

[10] Ratnesh Kumar, Edwin Weill, Farzin Aghdasi, and Parthsarathy Sriram. Vehicle re-identification: an efficient baseline using triplet embedding. *arXiv preprint arXiv:1901.01015*, 2019.

[11] Hongye Liu, Yonghong Tian, Yaowei Yang, Lu Pang, and Tiejun Huang. Deep relative distance learning: Tell the difference between similar vehicles. In *CVPR*, pages 2167–2175, 2016.

[12] Xinchen Liu, Wu Liu, Tao Mei, and Huadong Ma. A deep learning-based approach to progressive vehicle re-identification for urban surveillance. In *ECCV*, pages 869–884. Springer, 2016.

[13] Sashank J Reddi, Satyen Kale, and Sanjiv Kumar. On the convergence of adam and beyond. *arXiv preprint arXiv:1904.09237*, 2019.

[14] Ergys Ristani, Francesco Solera, Roger Zou, Rita Cucchiara, and Carlo Tomasi. Performance measures and a data set for multi-target, multi-camera tracking. In *ECCV*, pages 17–35. Springer, 2016.

[15] Ergys Ristani and Carlo Tomasi. Features for multi-target multi-camera tracking and re-identification. In *CVPR*, pages 6036–6046, 2018.

[16] Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jon Shlens, and Zbigniew Wojna. Rethinking the inception architecture for computer vision. In *CVPR*, pages 2818–2826, 2016.

[17] Zheng Tang and Jenq-Neng Hwang. Moana: An online learned adaptive appearance model for robust multiple object tracking in 3d. *IEEE Access*, 7:31934–31945, 2019.

[18] 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.

[19] 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 *CVPRW*, pages 108–115, 2018.

[20] Guanshuo Wang, Yufeng Yuan, Xiong Chen, Jiwei Li, and Xi Zhou. Learning discriminative features with multiple granularities for person re-identification. In *ACMM*, pages 274–282. ACM, 2018.

[21] Zhongdao Wang, Luming Tang, Xihui Liu, Zhuliang Yao, Shuai Yi, Jing Shao, Junjie Yan, Shengjin Wang, Hongsheng Li, and Xiaogang Wang. Orientation invariant feature embedding and spatial temporal regularization for vehicle re-identification. In *ICCV*, pages 379–387, 2017.

[22] Joe H Ward Jr. Hierarchical grouping to optimize an objective function. *Journal of the American statistical association*, 58(301):236–244, 1963.

[23] Nicolai Wojke, Alex Bewley, and Dietrich Paulus. Simple online and realtime tracking with a deep association metric. In *ICIP*, pages 3645–3649. IEEE, 2017.

[24] Chih-Wei Wu, Chih-Ting Liu, Cheng-En Chiang, Wei-Chih Tu, and Shao-Yi Chien. Vehicle re-identification with the space-time prior. In *CVPRW*, pages 121–128, 2018.

[25] Ke Yan, Yonghong Tian, Yaowei Wang, Wei Zeng, and Tiejun Huang. Exploiting multi-grain ranking constraints for precisely searching visually-similar vehicles. In *ICCV*, pages 562–570, 2017.

[26] Zhun Zhong, Liang Zheng, Donglin Cao, and Shaozi Li. Re-ranking person re-identification with k-reciprocal encoding. In *CVPR*, pages 1318–1327, 2017.

[27] Y Zhou, L Shao, and A Dhabi. Viewpoint-aware attentive multi-view inference for vehicle re-identification. In *CVPR*, volume 2, 2018.