Multi-Camera Vehicle Tracking with Powerful Visual Features and Spatial-Temporal Cue

Zhiqun He[1]*, Yu Lei[1]*, Shuai Bai[1]. Wei Wu[2].
[1]Beijing University of Posts and Telecommunications. [2]Nanjing University.
{he010103, 397680446, baishuai}@bupt.edu.cn, Wuwei.ai.bj@gmail.com

Abstract

Vehicle re-identification and multi-camera multi-object vehicle tracking are important components in the field of intelligent traffic, which is attracting more and more attention. In the NVIDIA AI City Challenge, we propose our solutions to solve these issues. In Track1 task, clustering loss and trajectory consistent loss are introduced into the vehicle re-identification training framework to train more suitable trajectory-based features for the clustering task. Besides, spatial-temporal cue is fully excavated to make up the deficiency of appearance feature and constrained hierarchical clustering is introduced into the pipeline to get the final cluster results. In Track2 task, we propose an effective vehicle training framework and trajectory-based weighted ranking method, which greatly improve the performance. Furthermore, an efficient way to mining the additional data to train more robust features is proposed to enlarge the training data. Finally, our algorithm achieves the state-of-the-art performance in the competition.

1. Introduction

Large scale traffic video analysis plays an important role in AI city, which is attracting more and more attention nowadays. Multi-camera multi-target vehicle tracking allows us to get the trajectory of the vehicles in the city. Much attention has been paid in recent years to the problem in vehicle re-identification(ReID) [16, 22, 7, 26]. Given a query, these works aim to make the query closer to the positive set than to the negative set so as to get a good ranking result. However, in MTMC task, the problem becomes how to cluster the trajectories across the cameras. In our work, we design a clustering loss to make the visual vehicle feature more suitable for the clustering task.

In the multi-camera multi-target pedestrian tracking, some works [14, 25, 21] has been done, which focus on how to find robust feature and effective cluster methods, without much consideration of the spatial-temportal cue. The AI city 2019 Track1 Challenge [19] provides the calibration of each camera for the participants, which can turn the pixel in the 2D image into latitude and longitude in the real world. In this work, we utilize this information to get the space-time similarity and to rerank the appearance similarity matrix, which further improved the accuracy of our algorithm.

In vehicle ReID, image-based similarity is used to get the ranking order of each query. However, in MTMC task, it becomes a trajectory-based matching problem. Many works only average all features within a single trajectory. Due to different postures of the vehicle in a trajectory, simply averaging the features may not be the best way. In our work, batch hard triplet loss proposed by [4] is introduced to make the query closer to the positive set in the the same camera than to positive set in different cameras. In this way, the features within a trajectory might have more consistency.

For Vehicle-ReID, image-to-image retrieval method cannot determine whether the images are in the same ID by their common areas due to the occlusion and blurring of corresponding images. However, when utilizing the retrieval method of image-to-track matching algorithm, there will be more chance to acquire abundant information from multiple perspectives with the track images, and thus help obtain more robust features for Vehicle-ReID. Based on the ideas above, we design an algorithm to acquire the quality of the track images without supervision by the outputs of the network, and assign appropriate weights to the features of the track with the results of the quality judgment.

In summary, we make the following contributions:

- The clustering loss and trajectory consistency loss are proposed to get powerful appearance vehicle features for MTMC task and trajectory-based clustering, which greatly improves the performance.
- The camera calibration information is fully used to turn the pixels position in the 2D images into latitude and longitude in the real world. After that, the spatial-temporal similarity can be generated as a complement to the appearance features. Besides, constrained hi-

*contributed equally

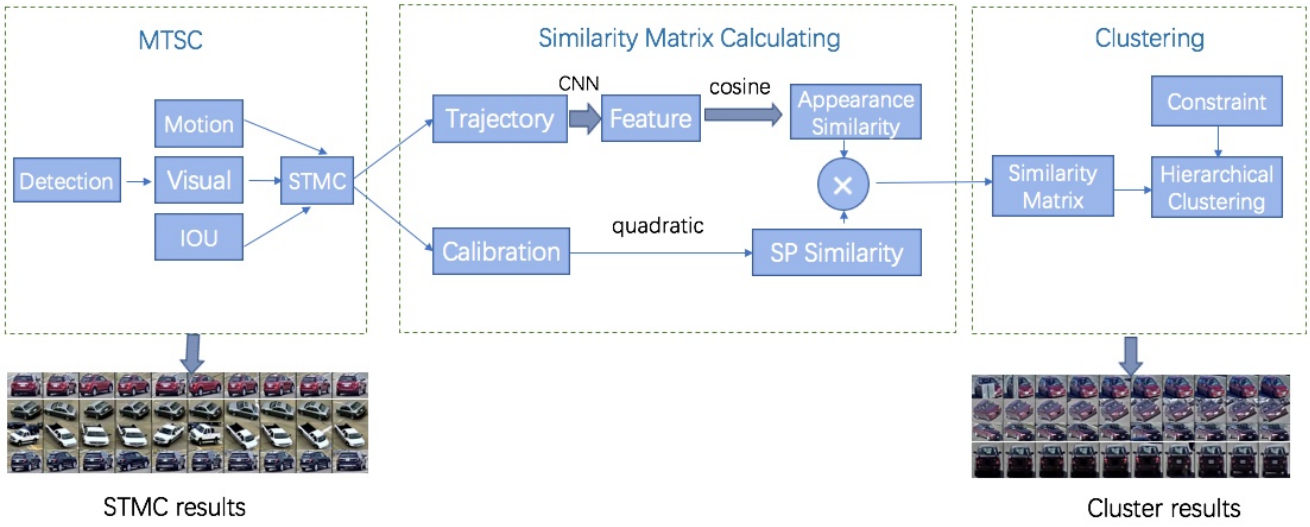

Figure 1. The overall design of our algorithm, which consists of MTSC module, similarity matrix calculating module and clustering module.

erarchical clustering algorithm is introduced into the algorithm.

- We propose a ranking method which fully utilizes the trajectory information to generate weighted feature for Vehicle ReID. An effective way of mining more adaptive data from the additional data is proposed to achieve a great performance.
- The methods are evaluated on the AI city 2019 Track1 and Track2 challenge and achieves the state-of-the-art performance.

2. Related Work

We summarize works on different aspects as follows.

Vehicle ReID. MTMC task relys on vehicle ReID to extract appearance feature of the vehicle. Liu .et al [8] proposes a well-annotated vehicle re-identification dataset, which contains 776 identifications with rich attitude. Shen. et al [16] proposes a two-stage framework for vehicle re-identification, which first proposes a series of candidate visual-spatial-temporal paths with the query images as the starting and ending states, then a Siamese-CNN+Path-LSTM network is utilized to make full use of spatial-temporal regularization from the candidate path. Wang. et al [22] extracts local region features of different orientations based on 20 key point locations to get the local details of the vehicle. Zapletal. et al [24] aligns images with 3D rectangular and use HOG as the feature extractor. Zhou. et al [26] proposes a viewpoint-aware attentive multi-view Inference model to solve the multi-view vehicle ReID problem.

MTSC. Tracking-by-detection becomes more and more popular for multiple objects tracking. So as to find the trajectory of each target from detection results in all frames, data association is an essential task, which is discussed in [12, 23, 18]. The linear programming and graph-based methods are used to conduct in a discrete space. Many optimization algorithms such as the network flow [15] and the subgraph multi-cut [5, 17] have been proposed.

MTMC. With the advancement of multiple objects tracking techniques and the vehicle ReID techniques, MTMC can be better solved. Ristani. et al [13] proposes a large-scale, well-annotated multi-camera tracking benchmark for pedestrians, which makes great progress in this field. Ristani et al [14] proposes an adaptive weighted triplet loss for training and a new technique for hard-identity mining. Zhang. et al [25] utilizes hierarchical clustering with well-trained person re-identification features in the DukeMTMC benchmark. Yonatan. et al [21] proposes a multiple hypothesis tracking algorithm for multi-target multi-camera tracking with disjoint views. In the NVIDIA AI City 2018 Challenge [11], Tang. et al [20] proposes a fusion of visual and semantic features for both single-camera tracking and inter-camera tracking. Feng. et al [2] focused on vehicle ReID and multiple object tracking to solve the MTMC task.

3. Proposed Method

3.1. Multi-Target Single-Camera(MTSC) Tracking

3.1.1 Detection

Lin. et al [7] proposes a novel feature pyramid network, which uses top-down and lateral connections so as to capture the objects with different scales. We use ResNet50

[3] as the backbone, which is trained in the training set of AI City Challenge Track1 and VisDrone2018 DET [27] dataset, with only the vehicle category. In the inference phase, the image is resized into 1440×800, so as to capture more small vehicles in the video. As the bounding boxes with the areas smaller than 1,000 pixels are not annotated, we filter them out in the result. Besides, we use the annotated ignored area provided by the competition organizing committee to further filter out the detection results. The confidence threshold of the detection is set to 0.7 to reduce false positive rectangles.

3.1.2 MTSC

Feng. et al [1] proposes a unified Multi-Object Tracking framework to make full use of long term and short term cues, which achieves state-of-the-art performance in the MOT benchmark [6]. Besides, the state-of-the-art single object tracking(SOT) method of [28] is used for capturing short term cues and a vehicle ReID method is applied to extracting long term cues. During data association process, motion information, location information and visual information are taken into consideration. Besides, potential switcher is used to make the association module more robust.

We use the detection algorithm mentioned above as the detector with high recall. Only the bounding boxes with the confidence score higher than 0.7 are taken into multiple target tracking algorithm. As the vehicles move fast in the dataset, we enlarge the search region to get more accurate locations. The MTSC results are shown in Figure 2.

Figure 2. Illustration of the MTSC results. The black area in the image is the annotated ignored area provided by the competition organizing committee.

3.2. Multi-Camera Vehicle Re-identificaiton Feature

3.2.1 Overall Design

We use the tricks proposed by [10], which achieves state-of-the-art results in the pedestrian re-identification field, to train our ReID feature. The training framework is shown in Figure3. The image is resized to 320×320 in the training and inference phase. We use ResNet50 as the backbone, with data augumentation such as random padding, horizontal flip and random erasing. For additional dataset, we experiment on VeRi-776[9] as a supplement.

In the inference phase, we generate a global feature with the dim of 2048 before batch normalization neck(BNNeck) as the final output of the input image. Similarity between two features is calculated through cosine similarity .The overall loss function contains the cluster loss, trajectory consistency loss and the classification loss, which will be discussed in detail in the next subsection.

3.2.2 Loss function

Cross Entropy Loss with Smooth Label. As there are many similar vehicles in the dataset, simply using hard label will make the training difficult. For each image, we generate soft label to encourage the model to train smoothly and less confidently, which can be expressed as:

$$q_i = \begin{cases} 1 - \frac{N-1}{N}\varepsilon, if\ y = i \\ \frac{\varepsilon}{N}, otherwise \end{cases} \tag{1}$$

where i is the index of the image, y is the identification of the image, N is the number of the samples in the dataset, and ε is a small constant, which is set to 0.1 in our work. Then the cross entropy loss with label smooth can be computed as:

$$L(ID) = \sum_{i=1}^{N} -q_i log(p_i) \tag{2}$$

where p_i is the ID prediction logits of class i.

Clustering Loss. ReID ranks distances to a query while MTMC classifies a pair of trajectories across cameras as being co-identical or not. Besides, the performance of them are measured by different metrics: ranking performance mAP for ReID, while IDF1 is used for MTMC, which suggest that appearance features used for the two problems must be learned with different kinds of loss functions. Specifically, the ReID loss ought to guarantee correct feature ranking for any given query while the MTMC loss should ensure that the largest distance between any two co-identical features is smaller that the smallest distance between any two non co-identical features, to obtain a margin between within-identity and between-identity distances. Therefore, we propose a clustering loss to ensure that every two samples in the same cluster centroid have smaller distance than that of any other two samples in different cluster centroids.

Alexander. et al [4] proposes batch hard triplet loss. Let $D_{i,j} = D(f_\theta(x_i), f_\theta(x_j))$, where f_θ is the function that

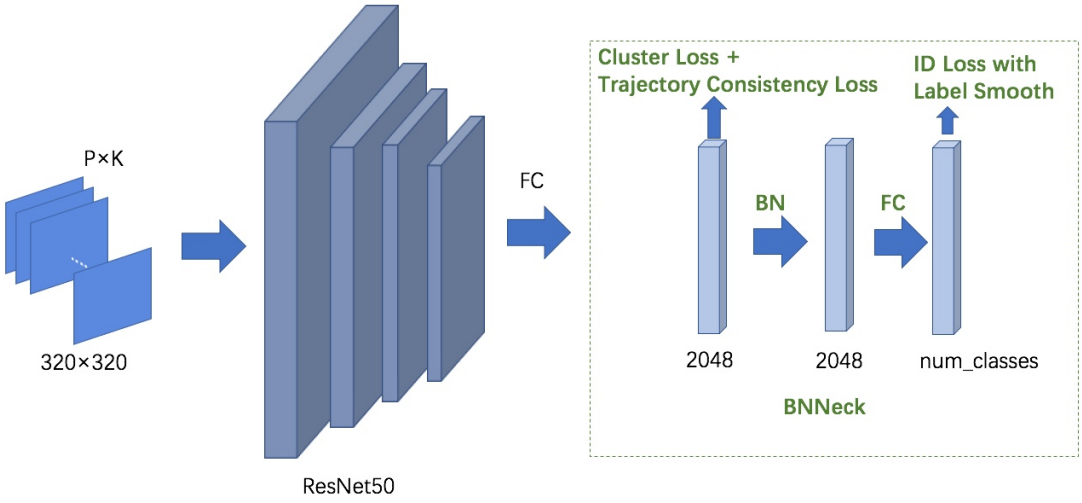

Figure 3. The overall design of the vehicle ReID algorithm, where P is the number of the ID in the mini batch and K is the number of images in each ID. The feature before BNNeck is used for cluster loss and trajectory consistent loss while the feature after BNNeck is fed into ID loss.

map the input x_i and x_j to the features of 2048 dimensions, i and j is the index of the samples. In the mini batch, let P be the number of identifications in the batch, K be the numbers of the images of each identification. For each anchor, make sure that the maximum distance of the positive pairs is larger than the minimum distance of the negative pairs by a margin m, which is set to 0.3 in our experiment.

$$L_{BH}(\theta; X) = \sum_{i=1}^{P}\sum_{a=1}^{K}[m + \max_{p=1,2,...,K} D(f_\theta(x_a^i), f_\theta(x_p^i)) - \min_{j=1,2,...,P,n=1,2,...,K,j\neq i} D(f_\theta(x_a^i), f_\theta(x_n^j))]_+ \tag{3}$$

where the data point x_j^i corresponds to the j-th image of the i-th identification in the batch, $[x]_+$ equals to $max(0, x)$.

In our work, we let the anchor be the cluster centroid, whose feature is the average features in the cluster. It can be expressed as:

$$L_C(\theta; X) = \sum_{i=1}^{P}[m + \max_{p=1,2,...,K} D(f_\theta(x_{a_i}^i), f_\theta(x_p^i)) - \min_{j=1,2,...,P,n=1,2,...,K,j\neq i} D(f_\theta(x_{a_i}^i), f_\theta(x_{a_n}^j))]_+ \tag{4}$$

where a_i is the cluster centroid of the identification with index i and a_n is the cluster centroid of the identification with index n.

Trajectory Consistent Loss. Through single camera tracking, we can get the trajectory under a single camera. Multi-camera tracking is actually a problem of clustering trajectories across different cameras. Our algorithm extracts the features of each image in the trajectory. Then we average the features of the trajectory to get robust visual discription. However, due to the variety of postures of the vehicle in some trajectory, feature fusion sometimes reduce the accuracy. Therefore, in the training phase, we utilize the batch hard triplet loss within the trajectory and identification. Therefore, the features in the trajectory are consistent, so that there is better fusion effect.

Let K equals to $C \times T$, where C is the number of the cameras in each identification and T is the number of images in the same trajectory. Then the trajectory consistent loss is:

$$L_{TC}(\theta; X) = \sum_{i=1}^{C}\sum_{a=1}^{K}[m + \max_{p=1,2,...,t} D(f_\theta(x_a^i), f_\theta(x_p^i)) - \min_{j=1,2,...,C,n=1,2,...,T,j\neq i} D(f_\theta(x_a^i), f_\theta(x_n^j))]_+ \tag{5}$$

The final loss in our experiment is:

$$L = \lambda_1 L_{ID} + \lambda_2 L_C + \lambda_3 L_{TC} \tag{6}$$

where λ_1 and λ_2 is set to 1 and λ_3 is set to 0.2 in our experiment.

The loss use in this work is shown in Figure 1.

3.3. Reranking with Spatial-temporal Cue

In the inference phase, all trajectories can be represented by a feature of 2048 dimensions. The appearance similarity

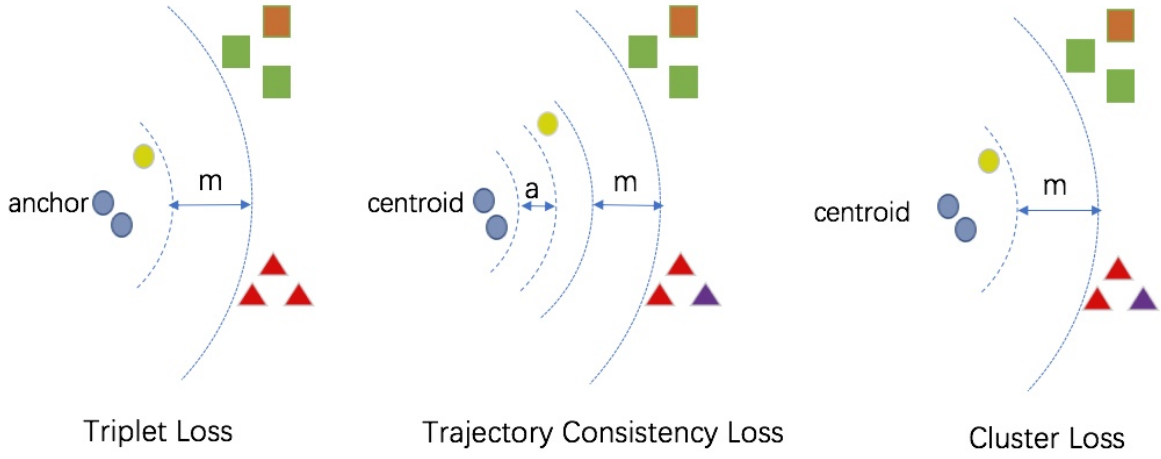

Figure 4. Illustration of different kinds of losses. Different shapes (triangle, circle, and square) represent different classes. The samples with the same color and the same shape are in the same trajectory.

of trajectory i and j can be computed using cosine similarity:

$$cos(i,j) = \frac{f_i \cdot f_j}{\|f_i\| \, \|f_j\|} \tag{7}$$

where f_i is the feature of i and f_j is the feature of j.

AI City 2019 Track1 Challenge provides camera calibration information, which can be used to map the coordinates (x, y) in the 2D image to the latitude and longitude (lat, lng) by the following equation in the real-world. Let $L = (x, y, 1)$, representing the position in the image. We select center point of the bottom edge of the vehicle as the (x, y) to calculate the distance.

$$P = H^{-1} \cdot L \tag{8}$$

P is a 1×3 vector. Then (lat, lng) can be computed as:

$$\begin{cases} lat = P[0]/P[2] \\ lng = P[1]/P[2] \end{cases} \tag{9}$$

After that, the distance in the real world between the two points can be calculated. We use spatial-temporal cue to calculate the similarity of the trajectories. Because there exists measurement error in the calibration, we use it cautiously. If there are time intersections between the two trajectories, the average distance d between them in the same time can be calculated. Then the average distance is mapped to the trajectory similarity p by the following quadratic function:

$$p_{i,j} = max(0, 1 - (d/200.)^2) \tag{10}$$

where i, j is the index of the two trajectories.

If the two trajectories have no time intersection, we calculate the average speed to pass from one camera to another camera. Then the average velocity v is mapped to the trajectory similarity p by the following quadratic function:

$$p_{i,j} = max(0, ((v - 10)/30.)^2) \tag{11}$$

All hyperparameters are calculated in the training set. The average speed pass from one camera to another camera is about $30m/s$ and the mean distance of the two same trajectories in the same time is about 0 meter. Finally, the similarity matrix S is $m \times m$, where m is the number of the trajectories. Each element of S is $S_{i,j} = p_{i,j} * cos(i, j)$.

3.4. Solving the Similarity Matrix with Constrained Hierarchical Clustering

The vehicles with the same identification contain a variety of angles. At the same time, there are many similar vehicles with different identification, which have large values in similarity matrix. Therefore, by using density clustering, such as the DBSCAN clustering, many similar vehicles with different ID will gather in the same class. Besides, because the number of the cluster centroid are not provided, it is not suitable to use k-means clustering.

While constructing the similarity matrix, the camera information is taken into consideration to rescore the appearance similarity matrix. Single camera tracking can get a accurate multi-target tracking result under a single camera. Therefore, if the two tracks are under the same camera, they are considered to be impossible to be the same car. This

constraint reduces the complexity of the problem. As cluster loss and trajectory consistency loss are utilized to train the vehicle ReID model, the appearance feature of the trajectories are discriminable. Therefore, we solve this problem by hierarchical clustering with constraints. The two trajectories with the largest similarity value in the matrix are most likely to have the same identification.

- Step1. Initially, treat each sample as a cluster;
- Step2. Calculate the similarity between each cluster;
- Step3. Assign the similarity of the two cluster under the same camera to 0;
- Step4. Find the two most recent clusters and classify them into one cluster centroid;
- Step5. Repeat step 2, step 3 and step 4 until the similarity between each two samples is not greater than the threshold (in this dataset, the threshold is set to 0.65.)

4. Vehicle ReID

4.1. Ranking with weighted features and trajectory infomation

When utilizing ReID in traffic monitoring, it may require us to replace the single image with a brief trajectory as there may not be abundant information in a single image and some features may also be obscured. While with the trajectory that covers multiple similar images, the features of the same target can be gathered from different perspectives. Inspired by the ideas above, different from the traditional solution that compares the features between the query images and gallery images, we can replace the features of each gallery image with the average features of the track that this gallery image lie in, which can be regarded as an improvement from image-to-image to image-to-track way.

However, there still remains some challenges in image-to-track ReID. For example, the images in a same trajectory may be in different quality caused by the target obscured or blurring. And the images in poor quality cannot represent the features of the trajectory. To address this problem, the most direct solution is to judge the quality of the images with extra models, and assign an appropriate weight to each image by the results of the quality judgment. While considering that this solution may call for additional labelling costs, seeking for an unsupervised approach to tackle this challenge may be a valuable and popular choice.

Intuitively, considering the query that has the same id as the trajectory, the higher the quality of the trajectory image is, the more similar they are. And thus we can select the images from the query which are the most similar to the trajectory to get a weighted vector of this trajectory by the distance matrix between it and the selected query.

Specifically, at first, we calculate the distance matrix $Dist(Q, G)$, where Q is the image set of query, denoted as $\{Q_1, Q_2, ...Q_m\}$, G is the image set of gallery, denoted as $\{G_1, G_2, ...G_m\}$. Then we can calculate the sub-matrix $Dist(Q, T_i)$, where T is the set of trajectory from gallery, and T_i is the i^{th} trajectory in T. Then we choose the rows of $Dist(Q, T_i)$ whose min values are lower than 0.2, denoted as $D^{'}$, to get the most similar images as the trajectory. Then We calculate the mean value of each column in $D^{'}$ to get an average distance vector A_i of T_i. The weighted vector can be calculated as the following equation.

$$W_{ij} = \frac{1}{A_{ij} + 0.01} \tag{12}$$

With the weighted vector, we can calculate the weighted average feature of the track as shown in the follow equation

$$f_{t_i} = \sum\nolimits_{j=0}^{s_i} F(T_i)_j * W_j \tag{13}$$

Where f_{t_i} is the weighted features of T_i, and $F(T_i)$ is the feature set of T_i. Then we can get the image-to-track distance matrix $Dist(Q, t)$ to get the ReID result and tile tracks' images.

4.2. Mining additional dataset to domain adaption

Data is vital for deep learning tasks, and in general, additional data from the same domain will contribute to a great improvement of the performance, while as for the data from different domains, the effect for the performance is doubtful. The reasons may lie in that additional data can dilute the distribution of the original dataset to some extent and make the distribution of the prediction results closer to the additional datasets. To tackle this problem, we utilize the model trained by the original dataset to generate the features of additional datasets. With the newly generated features as gallery and the features of the original datasets as query, we can retrieve the data that is closer to the original dataset. As shown in the experimental results, the ranking results from other datasets by the training data of AI City 2019 track2 can really match well in appearance. For additional dataset, we experiment on VeRi-776[9] and our own datasets.

5. Experiment

5.1. Implementation Details

Our algorithm is implemented in PyTorch 1.0.1. The experiments are performed on four GeForce GTX TITAN XP GPUs. In the vehicle ReID training process, we use ResNet50 as the backbone. In the first 10 epoch, we use the warmup operation to initialize the network with a small learning rate. We perform data augumentation by random erasing and random padding. Label smoothing is used to generate the soft label. As pointed by [10], BNNeck is used to perform BN operations after the final fully connected layer and then the feature is used for ID loss. In the inference phase, the feature before BNNeck is used for better

performance. L2 normalization is used to normalize the feature of each vehicle. We set the similarity threshold to 0.7. The similarity between two trajectories is not considered to be the same identification. We train the vehicle ReID model in the Track2 dataset with 73 IDs among 333 IDs selected as the validation set and we use top100 mean average precision(mAP) to evaluate the performance. The performance of the MTMC task is evaluated in the training set in Track1. The results of the vehicle ReID model in the validation can be seen in Figure 5.

Figure 5. Illustration of vehicle ReID model. The first image is the query image and the other ten images are the 10-nearest neighbors. Green and blue box correspond to the positives and negatives, respectively.

5.2. Ablation Analyses in Track1

Influence of MTSC methods. We compare our MTSC method with DeepSort[23], MOANA[18], TC[20], which is shown in Table 1. The introduced MTSC methods achieves the better performance in our experiments because the introduced MTSC methods has higher recall.

Influence of Clustering Methods. Besides, we compare our constrained hierarchical clustering with DBSCAN and K-means. As can be seen in Table 1, our proposed method is far ahead of DBSCAN and K-means.

Influence of Loss Functions. In the experiment, we try different combination of different loss functions, as shown in Table 1. By adding cluster loss, the mAP will decrease with higher IDF1, because cluster loss is more suitable for MTMC task. The trajectory consistent loss contribute to both mAP and IDF1. In the table, we come to the conclusion that trajectory feature fusion can boost the performance in MTMC task.

Influence of Spatial-Temporal Cue. In table1, we evaluate the importance of spatial-temporal cue. When we only use the baseline methods, the feature of the vehicle is not so powerful. Therefore by introducing the spatial-temporal cue, it gains a 1.1% improvement in IDF1. When the trajectory feature fusion and effective loss functions are utilized, the performance improvement brought by spatial-temporal cue is minor

AI City Track1 MTMC results				
	mAP	IDF1	IDP	IDR
DeepSORT+C+TC+SP+TF+CHC	0.635	0.693	0.684	0.703
TC+C+TC+SP+TF+CHC	0.635	0.715	0.706	0.725
MOANA+C+TC+SP+TF+CHC	0.635	0.680	0.669	0.692
Baseline(B) + CHC	0.641	0.757	0.734	0.782
B + SP + CHC	0.641	0.768	0.766	0.772
B+TF+CHC	0.641	0.803	0.771	0.839
B+TF+DBSCAN	0.641	0.598	0.593	0.604
B+TF+K-means	0.641	0.497	0.502	0.494
B+C+TF+CHC	0.631	0.812	0.785	0.841
B+C+TC+TF+CHC	0.635	0.819	0.791	0.849
B+C+TC+SP+TF+CHC	0.635	0.826	0.801	0.853
B+C+TC+SP+TF+CHC(TestSet)	**0.730**	**0.665**	**0.693**	**0.640**

Table 1. The table shows the mAP in vehicle ReID validation set and IDF1, IDP, IDR in Track1 training set, where Baseline uses only cross entropy and batch hard triplet loss, C represents the cluster loss function, TC represents the trajectory consistent loss function, SP represents the spatial-temporal cue, TF represents trajectory feature fusion, CHC represents constrained hierarchical clustering.

5.3. Ablation Analyses in Track2

Influences of the Number of Batch Size. The mini-batch of triplet loss contains $B = P * K$, where P and K denote the number of different vehicles and the number of different images per vehicle, respectively. The experiment result is shown in Table 6. As can be seen, $P = 16, K = 8$ achieves the best performance.

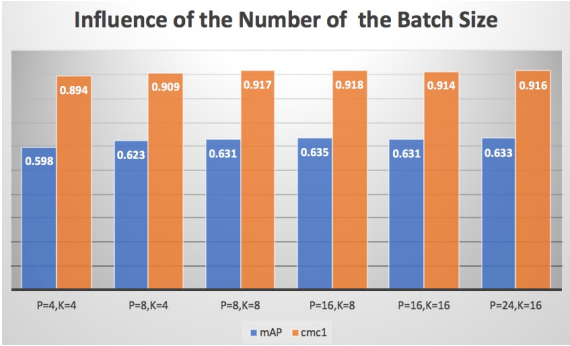

Figure 6. The figure shows the mAP and cmc1 in the validation in Track2 for different image size.

Influences of Image Size. We trained models without center loss and set $P = 16$, $K = 8$ in 4 GPUs with 32 images per GPU. As shown in Figure 7, the image size is a pretty importance factor for the performance of ReID models. When the input size of the image becomes larger within a certain range, the performance become better. The input size of 320×320 achieves best performance in the experiment.

Influence of Different Ranking Method. In the experiment, we try different ranking methods on the validation

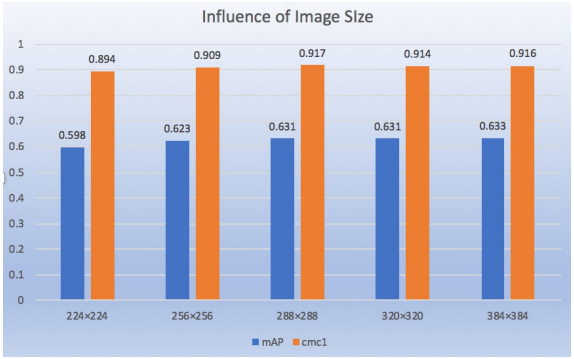

Figure 7. The figure shows the mAP and cmc1 in the validation in Track2 for different image size.

Ranking Method	mAP	cmc1
I2I(+rr)	0.635(0.7232)	0.918(0.9232)
I2T+average feature(+rr)	0.6941(0.7417)	0.9321(0.9238)
I2I+weighted feature(+rr)	0.7315(0.7503)	0.9235(0.9385)

Table 2. The table shows the mAP and cmc1 in the validation in Track2 for different ranking method, where rr represents re-ranking method, I2I represents image-to-image, I2T represents image-to-track.)

Data-fusion Method	mAP	cmc1
A	0.585	0.872
A+V(1/2)	0.630	0.904
A+V(all)	0.635	0.918
A+V(mining 1/2)	0.640	0.925
A+V(all)+O(all)	0.654	0.930
A+V(mining)+O(mining)	0.670	0.942

Table 3. The table shows the mAP and cmc1 in the validation in Track2 for different ranking method.A represent AICITY Track2 training data, V represents VeRi-776 dataset, O represents our own datasets

set, and the results as shown in Table 2. In these methods, we all use re-ranking to achieve a better performance. By adding the track info and calculating the average features of the track, the mAP can improve by 1.9%. And it can make further improvement by the weighted features generated from our algorithm.

Influence of Different Data-fusion Methods. For data-fusion, we attempt two methods: adding all additional data and adding mining data by the method of section 4.2. The experimental results are shown in Table 3. There is no improvement than without additional data for the former method. By mining adaptive data, the mAP can improve the performance by nearly 1%.

5.4. Performance Evaluation of Challenge Contest.

Here we (team ID 12) report our challenge contest performance of the two tracks: City-Scale Multi-Camera Vehicle Tracking(Track1) and City-Scale Multi-Camera Vehicle Re-Identification(Track2).

AI City Track2 ReID Results		
Method	mAP	cmc1
MoVI+BH[19]	0.265	0.484
PCB + additional data + rr	0.674	0.754
MGN + additional data + rr	0.691	0.765
Ours+rr	0.730	0.816

Table 4. The table shows the results of different method on Track2 Test Set, where rr represents re-ranking.

Performance evaluation of Track1		
Team ID	IDF1	Rank
21	0.7059	1
49	0.6865	2
12(Ours)	**0.6653**	**3**
53	0.6644	4
97	0.6519	5
59	0.5987	6
36	0.4924	7
107	0.4504	8

Table 5. Performance evaluation of challenge contest in Track1.

Performance Evaluation of Track2		
Team ID	rank100-mAP	Rank
59	0.8554	1
21	0.7917	2
97	0.7589	3
4	0.7560	4
12(Ours)	**0.7302**	**5**
53	0.6793	6
131	0.6091	7
5	0.6078	8

Table 6. Performance Evaluation of challenge contest in Track2.

In Track1, our IDF1 score is 0.6653, which ranks number 3 in the overall evaluation. In Track2, we rank number 5 among all the teams with the mAP of 0.7302, as can been in Tabel 4. The Performance evaluation of Track1 and Track2 are shown in Table5, Table6, respectively.

6. Conclusion

In Track1, we fully utilize clustering loss and trajectory consistency loss to get powerful visual vehicle features for MTMC task. Trajectory-based features are used to generate the appearance similarity matrix. Spatial-temporal cue is excavated to rescore the appearance similarity matrix, as a supplement of the appearance feature. After that, we use hierarchical clustering with camera constraints to obtain the cluster results of all the trajectories. In Track2, a image-to-track ReID ranking method with weighted feature is proposed to capture more temporal information in the trajectory. Besides, a data mining methods is utilized to help the training more stable and effective.

References

[1] Weitao Feng, Zhihao Hu, Wei Wu, Junjie Yan, and Wanli Ouyang. Multi-object tracking with multiple cues and switcher-aware classification. *CoRR*, abs/1901.06129, 2019.

[2] Weitao Feng, Deyi Ji, Yiru Wang, Shuorong Chang, Hansheng Ren, and Weihao Gan. Challenges on large scale surveillance video analysis. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, June 2018.

[3] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2016.

[4] Alexander Hermans, Lucas Beyer, and Bastian Leibe. In defense of the triplet loss for person re-identification. *CoRR*, abs/1703.07737, 2017.

[5] Margret Keuper, Siyu Tang, Zhongjie Yu, Bjoern Andres, Thomas Brox, and Bernt Schiele. A multi-cut formulation for joint segmentation and tracking of multiple objects. *CoRR*, abs/1607.06317, 2016.

[6] Laura Leal-Taixé, Anton Milan, Ian D. Reid, Stefan Roth, and Konrad Schindler. Motchallenge 2015: Towards a benchmark for multi-target tracking. *CoRR*, abs/1504.01942, 2015.

[7] Tsung-Yi Lin, Piotr Dollar, Ross Girshick, Kaiming He, Bharath Hariharan, and Serge Belongie. Feature pyramid networks for object detection. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, July 2017.

[8] X. Liu, W. Liu, H. Ma, and H. Fu. Large-scale vehicle re-identification in urban surveillance videos. In *2016 IEEE International Conference on Multimedia and Expo (ICME)*, pages 1–6, July 2016.

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

[10] Hao Luo, Youzhi Gu, Xingyu Liao, Shenqi Lai, and Wei Jiang. Bag of tricks and A strong baseline for deep person re-identification. *CoRR*, abs/1903.07071, 2019.

[11] Milind Naphade, Ming-Ching Chang, Anuj Sharma, David C. Anastasiu, Vamsi Jagarlamudi, Pranamesh Chakraborty, Tingting Huang, Shuo Wang, Ming-Yu Liu, Rama Chellappa, Jenq-Neng Hwang, and Siwei Lyu. The 2018 nvidia ai city challenge. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, June 2018.

[12] R. Nevatia. Global data association for multi-object tracking using network flows. In *2008 IEEE Conference on Computer Vision and Pattern Recognition*, pages 1–8, June 2008.

[13] Ergys Ristani, Francesco Solera, Roger Zou, Rita Cucchiara, and Carlo Tomasi. Performance measures and a data set formulti-target, multi-camera tracking. In Gang Hua and Hervé Jégou, editors, *Computer Vision – ECCV 2016 Workshops*, pages 17–35, Cham, 2016. Springer International Publishing.

[14] Ergys Ristani and Carlo Tomasi. Features for multi-target multi-camera tracking and re-identification. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2018.

[15] Samuel Schulter, Paul Vernaza, Wongun Choi, and Manmohan Chandraker. Deep network flow for multi-object tracking. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, July 2017.

[16] Yantao Shen, Tong Xiao, Hongsheng Li, Shuai Yi, and Xiaogang Wang. Learning deep neural networks for vehicle re-id with visual-spatio-temporal path proposals. In *The IEEE International Conference on Computer Vision (ICCV)*, Oct 2017.

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

[18] Z. Tang and J. Hwang. Moana: An online learned adaptive appearance model for robust multiple object tracking in 3d. *IEEE Access*, 7:31934–31945, 2019.

[19] 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. *CoRR*, abs/1903.09254, 2019.

[20] 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 *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, June 2018.

[21] Yonatan Tariku Tesfaye, Eyasu Zemene, Andrea Prati, Marcello Pelillo, and Mubarak Shah. Multi-target tracking in multiple non-overlapping cameras using constrained dominant sets. *CoRR*, abs/1706.06196, 2017.

[22] 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 *The IEEE International Conference on Computer Vision (ICCV)*, Oct 2017.

[23] N. Wojke, A. Bewley, and D. Paulus. Simple online and realtime tracking with a deep association metric. In *2017 IEEE International Conference on Image Processing (ICIP)*, pages 3645–3649, Sep. 2017.

[24] Dominik Zapletal and Adam Herout. Vehicle re-identification for automatic video traffic surveillance. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, June 2016.

[25] Zhimeng Zhang, Jianan Wu, Xuan Zhang, and Chi Zhang. Multi-target, multi-camera tracking by hierarchical clustering: Recent progress on dukemtmc project. *CoRR*, abs/1712.09531, 2017.

[26] Yi Zhou and Ling Shao. Viewpoint-aware attentive multi-view inference for vehicle re-identification. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2018.

[27] Pengfei Zhu, Longyin Wen, and Dawei Du. Visdrone-det2018: The vision meets drone object detection in image challenge results. In *The European Conference on Computer Vision (ECCV) Workshops*, September 2018.

[28] Zheng Zhu, Qiang Wang, Bo Li, Wei Wu, Junjie Yan, and Weiming Hu. Distractor-aware siamese networks for visual object tracking. In *The European Conference on Computer Vision (ECCV)*, September 2018.