Pedestrian Tracking through Coordinated Mining of Multiple Moving Cameras

Yanting Zhang
Donghua University
Shanghai, China
ytzhang@dhu.edu.cn

Qingxiang Wang
Donghua University
Shanghai, China
170120118@mail.dhu.edu.cn

Abstract

Multiple object tracking has attracted great interest in the computer vision community. Most researchers focus on the applications under a single static or moving camera. More recently, tracking across multiple static cameras is also investigated due to the need for surveillance purposes. With the growing development of autonomous driving, it is critical to correlate all the vehicles' vision systems on the road to achieve a global perception. However, tracking across multiple moving cameras has not been well studied yet. We observe a lack of such a publicly available dataset for coordinated mining of multiple moving cameras. In this paper, we aim to bridge the gap and propose a new dataset of multiple moving cameras, called "DHU-MTMMC", in which the videos are collected from several cameras mounted on the moving cars. The dataset contains fourteen sequences in different scenarios with annotated pedestrians. We propose a baseline MTMMC workflow to deal with tracking pedestrians across cameras. When the joint detection and embedding are performed, the association algorithm can run online under single-camera settings. We treat multi-camera tracking as a linear assignment problem that can be solved efficiently. The overall IDF1 of the proposed MTMMC tracking on the dataset is 57.8%.

1. Introduction

Autonomous vehicles are gradually available in people's daily lives. The comprehensive and timely perception of road condition is of great significance to improve the driving safety. Currently, there are still many challenges in designing an complete autonomous driving perception systems [39]. Considering that the visual sensors from on-road moving vehicles record a large number of video data, it is meaningful to coordinatedly mine the information from multiple cameras to get a comprehensive knowledge about the environment. Benefiting from the rapid development of mobile communication technology, 5G or 5G+ will provide a good V2X communication to allow all synchronized cameras share the analyzed results, which can further assist the intelligent driving of on-road vehicles. Thus, it is a quite valuable attempt to take a further step into fusing the multi-source video data, rather than dealing with only a single-source data.

Tracking is a major task in computer vision field, especially when talking about autonomous cars. Reliably tracking the pedestrians on the road can help with future tasks like trajectory prediction and route planning for the autonomous car. This paper investigates the tracking issues for multiple pedestrian tracking under multiple cameras. Multi-Target Multi-Camera Tracking (MTMCT), which has gained growing attention in the community [32, 15], aims to locate every detected object at all times from videos taken by multiple cameras. Many remarkable works have been reported, including indoor [25, 33] and outdoor scenarios [15], enabling a wide range of applications such as visual surveillance, suspicious activity, and anomaly detection. Sport-player tracking and crowd behavior analysis can also be related to MTMCT tasks.

Currently, the cameras evolved in the MTMCT are usually installed at fixed locations [9, 1]. The relationship and connectivity among these static cameras can be easily established. But the flexibility of camera views is reduced. MTMCT under multiple moving cameras is a notoriously difficult problem [21]. More specifically, the spatial relationship among cameras is unknown, and the fields of views change all the time. Even though the subject is challenging, it has good scientific value. The research results are expected to overcome the limitations of the existing tracking framework and to achieve global perception in complex traffic scenarios.

When it comes to the approaches for tracking, previous researches have mainly focused on a single moving camera [40, 27] or multiple static cameras [29, 25]. The improved detection and feature embedding are expected to lift the tracking performance to a large extent [5]. The

MTMCT sometimes can be regarded as a Re-ID problem [41, 10, 26, 34] with additional information from spatial/temporal/appearance cues. In the static multi-camera settings, time constraints and connected exit/entry zones can be exploited with a camera link model [15]. Considering that a driving recorder or other visual sensor on the vehicle passively records the dynamic scene data with the vehicle enrolled in the traffic, the visual data from cameras on different vehicles can be substantially explored for a dynamic surveillance. The tracking can be much more challenging due to the changes from different lighting conditions, viewing perspectives, and scene differences, compared with traditional tracking across multiple static cameras. However, sharing information between on-road vehicles is beneficial for a global and accurate perception of the driving environment. We believe in the future development of autonomous driving and the construction of smart roads, information will be more mixed. The information from coordinated mining will be shared and give insights into path planning, obstacles avoidance, and so on. Consequently, systematic understanding of visual data collected from different moving cameras becomes highly demanding, which brings computer vision to our daily life more and more closely.

To sum up, in view of the limitations of the existing tracking frameworks and related visual sources, in this paper, multi-source video data are introduced and we study the key technologies involved in multiple pedestrian tracking by coordinated mining of multiple moving cameras. The contributions of this work are as follows.

(1) We propose a multi-target and multi-moving camera dataset, called "DHU-MTMMC", which is collected for multiple object tracking across different moving cameras. It bridges the gap between the increasing need for correlating moving vehicles on the road and lacking of such a dataset in the community.

(2) We carry out a joint object detection and embedding extraction, and use the Hungarian algorithm for single camera based tracking. We explore to use the Jonker Volgenant algorithm for tracklets assignment across cameras. It is simple but effective for association.

(3) A detailed comparison of single camera tracking based algorithms is presented to pave the way for our investigation of the more challenging multi-camera based solutions.

The rest of the paper is organized as follows. Section 2 gives a brief survey on the related work. In Section 3, the details of our proposed dataset is provided. Section 4 depicts the methodologies used in the proposed framework. The experimental results are shown in Section 5, followed by the conclusion in Section 6.

2. Related Work

In the recent past, the computer vision community has witnessed a tremendous development in object tracking, which can be pretty challenging especially with multiple objects involved. In this section, we will give a literature review in this field from the aspects of single camera based and multiple camera based datasets and methods for multiple object tracking.

2.1. Single Camera

It's meaningful to improve the performance on multiple objects tracking under single cameras. There are a variety of scenarios which are based on the tracking results, for example, crowd counting, flow analysis, and anomaly detection on highway or within a building. These tasks are highly coupled and interconnected, based on the detection and tracking. PETS dataset [12] is an early benchmark for multiple object tracking, targeted primarily at surveillance applications. It consists of several subsets, including people tracking.

Nowadays, MOTChallenge [8], offering a collection of datasets, has been a basis for the fair evaluation of multiple object tracking (MOT) algorithms. Abundant pioneer works have been proposed to improve the tracking performance [37, 40], and MOT dataset provides a public recognized platform for fair comparison. Recently, the KITTI benchmark [13] has been introduced to solve several hot topics in autonomous driving, such as odometry, object detection and orientation estimation, as well as tracking. Drones equipped with cameras have also been deployed to a wide range of applications, including aerial photography, delivery, and surveillance. The VisDrone challenge [42] has presented a benchmark for various important computer vision tasks, including detection, single-object tracking, multi-object tracking, and crowd counting.

These datasets have provided a convenience for researchers to develop their algorithms. There are several directions for tracking approaches. The most used methods are based on tracking-by-detection schemes [38, 6, 22, 19], i.e., we first detect objects, based on which we extract the corresponding features and then do the tracking. Previously, color and texture information are leveraged to discover the potential objects. Traditional feature extraction methods like color histograms, HOG, and LBP are utilized for describing the objects. Recently, people gradually choose to use deep learning based methods like Faster R-CNN [30] and YOLO [4] for object detection, followed by a step of extracting deep convolutional neural network (CNN) features for representation. As far as the representations are available, the tracking among adjacent frames can be solved benefiting from the similarity or motion information. Some researchers treat this problem as a graph model [32], where each detection is regarded as a node, and the correlation val-

ues constitute the edges among nodes. The tracking problem is solved by minimizing the total cost. Similarly, some researchers [36] propose to use tracklets as nodes and focus on the long-term tracking using graph-based model. In these above mentioned approaches, detection and embedding are separately dealt with, resulting in high computation complexity and redundancy. Some researchers [37, 5] propose joint detection and embedding workflows which can save the computation resources and lift the efficiency to a large extent, because the parameters can be reused both on detection and embedding. In this paper, we also follow their ideas, and carry out the detection and embedding in a single forward pass.

There are also some work which tries to output tracking results directly, e.g., Recurrent neural network (RNN) -based tracking [28]. However, the relevance for two faraway detections is almost lost, thus, the performance of RNN-based methods usually degrades in the long run. Feichtenhofer et al. [11] proposes simultaneous detection and tracking, using a multi-task objective for frame-based object detection and across-frame track regression and link the frame level detections based on across-frame tracklets to produce high accuracy detections at the video level. However, huge training data is needed for this kind of end-to-end tracking frameworks. Kang et al. [17] proposes a Tublet proposal network by using the object detection in the first frame to align a tube sequence. The result will not be satisfied when the objects exhibit a relatively large motion in the images, e.g., in highway scenarios.

2.2. Multiple Cameras

Benefiting from the vast advancement from single camera based tracking, people try to pursue a better performance in multi-camera scenarios. Nowadays, a vast number of cameras has been observed in traffic hubs and shopping centers. An automated MTMC tracking will be helpful through analyzing video streams taken by multiple cameras. Ristani et al. [31] propose a DukeMTMCT benchmark, a large-scale tracking dataset with $2.8k$ identities though no longer publicly available. They also design a method [32] based on correlation clustering. Learning good correlations is proven to make training simpler and less expensive. AI City Challenges [29] pave the way to get actionable insights derived from data captured by sensors in transportation. It performs vehicle re-identification based on vehicle crops from multiple cameras placed at multiple intersections. It also tracks vehicles across multiple cameras both at a single intersection and across multiple intersections. Spatial-temporal relations between cameras are informative to be considered [23, 35]. Due to the movement of vehicles usually follow certain driving patterns based on road structures and traffic rules, Hsu et al. [15] group them into limited numbers of trajectories, and propose the trajectory-based camera link models for multi-camera tracking of vehicles. With a reliable camera link model, the candidate set for matching becomes much smaller. As a result, the accuracy of across camera association can be significantly improved. Styles et al. [33] propose a dataset of multi-camera pedestrian trajectories from a network of 15 synchronized cameras set up indoors and forecast the future trajectory of an object across multiple non-overlapping camera views. Marroquin et al. [25] also introduce an indoor multi-camera multi-space dataset.

The aforementioned datasets are all under the settings of multiple static cameras. Interestingly, there has been rather limited work on the standardization of multiple target tracking across moving cameras. It can be much more challenging when carrying out the tracking under the settings of multiple moving cameras. Lee et al. [21] propose a framework to track on-road pedestrians across multiple driving recorders. Though the idea is similar to ours, but the dataset is not publicly available and we believe the deep learning based detection and embedding can boost their performances to a large extent. A public dataset is helpful to advance the state-of-the-art in the respective research fields. In the future, we are going to release the dataset. Moreover, we will keep enlarging the dataset to increase the difficulty by including more sequences filmed from different road types, with different weather/lighting conditions, and far more crowded scenarios. With this benchmark we would like to pave the way towards a more mature multi-target tracking system under multiple moving cameras.

3. Dataset

To the best of our knowledge, there is no public dataset for object tracking under multiple moving cameras. Previous datasets for cross camera tracking are mostly based on the static cameras. Figure 1 shows an overview of our proposed dataset, DHU-MTMMC, aiming at multi-target multi-moving camera tracking.

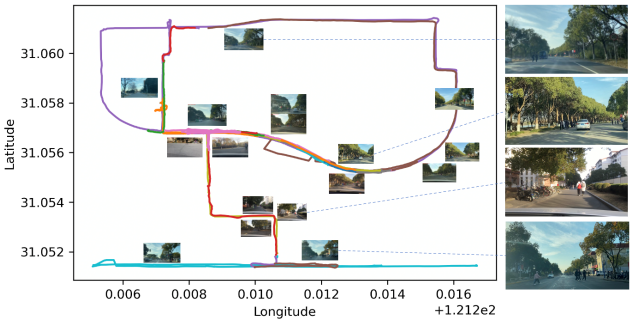

Figure 1. Overview of the proposed dataset: DHU-MTMMC. Different colors in the left represent different drives on the road to record videos. Snapshot images in the right show exampled different road scenarios.

Table 1. Configurations of the devices

Device	Type	Resolution	fps
1	Iphone 6S	1920 × 1080	30
2	Iphone 11	1920 × 1080	30
3	Iphone 8	1920 × 1080	30
4	Oppo Reno3	1920 × 1080	30

To collect the dataset, we have leveraged the cell phones due to its high resolution. We can also install other software like "SensorLog" in the cell phone to record Global Positioning System (GPS) locations as references. The configurations of the devices are shown in Table 1. The images are captured at the resolution of 1920×1080 and a frame rate of 30 fps. We fix each cell phone with a car. Then several cars carrying the cameras are driven within a campus to record live videos. Several different driving cases are considered, as is shown in Figure 2. Two vehicles may (a) move towards each other or (b) move in the same direction one-by-one. Three vehicles have an overlapping view, as is shown in Figure 2(c). Note that, the precise timestamp for the videos can be resolved through analyzing the video streams. The recorded videos are pre-sychronized.

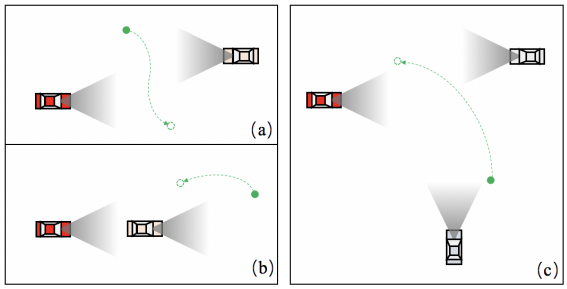

Figure 2. Different driving cases considered during the data collection. Some possible exampled pedestrian movements in green color are also shown.

As shown in Figure1, multiple drives from different vehicles are carried out in the campus. The trajectories are plotted based on the GPS cues. The cameras mounted on the cars observe plenty of pedestrians on the road. We have selected 6 scenes from all the drives, denoted as A, B, C, D, E, and F. Each scene covers the interaction from two or three cameras on different cars. We have manually annotated the pedestrians appeared in the videos as ground truth, following the same format with MOT dataset. Some of the sequences include crowded pedestrian crossings, making the dataset quite challenging. Since the vehicles move in a relatively slow speed when recording data, thus we annotate the videos at a frame rate of 5 fps to reduce redundancy and keep enough disparity among adjacent frames. We show the basic statistic information of these scenes in Table 2.

Table 2. Overview of the datasets

Sequence	Device	Length	Tracks	Boxes	Density
A-I	2	14s	4	171	1.9
A-II	1	52s	3	299	1.15
B-I	2	17s	23	837	7.27
B-II	1	21s	34	1041	9.91
C-I	2	9s	6	99	2.2
C-II	1	16s	16	880	11
D-I	2	84s	28	1262	3
D-II	1	86s	33	1598	3.7
E-I	2	30s	7	590	3.9
E-II	4	30s	2	148	0.98
E-III	3	25s	7	738	5.9
F-I	2	14s	5	186	2.65
F-II	4	12s	8	337	5.61
F-III	3	12s	4	185	3.08

For example, scene A contains two sequences of A-I and A-II captured by Device 1 and 2 from two cars. The video lengths for these two video sequences are 14s and 56s, respectively. Sequence A-I contains 4 tracks with 171 boxes, while A-II has 3 tracks with 299 boxes. The column of "Density" represents the average of the number of pedestrian per image.

Different scenarios are considered to demonstrate the performance of the proposed method. Due to the light reflection, the images can be pretty different under different video sources (we refer the readers to check with Figure 5 in Section V). In scene A, two vehicles move towards each other, thus, these two cameras have overlapping views for a while. Especially, a girl in yellow color is observed by two cameras at some time, then, she disappears in the first camera and later comes back in the field of view of the same camera. In scene B, two vehicles move one-by-one along the same direction, thus, pedestrians may walk out of the field of view of one camera and then enter into the other camera's field of view. In scene C and D, the vehicle turns around a corner, that is, pedestrians are likely to appear again in the camera's field of view after leaving it for a while. In scene E and F, three cameras have overlapping views, i.e., some people are simultaneously observed by these cameras.

4. Method

The whole workflow for the multi-target multi-moving camera (MTMMC) tracking is shown in Figure 3, which is composed of three parts, including joint detection and embedding (JDE), single camera based tracking, and multi-camera based tracking.

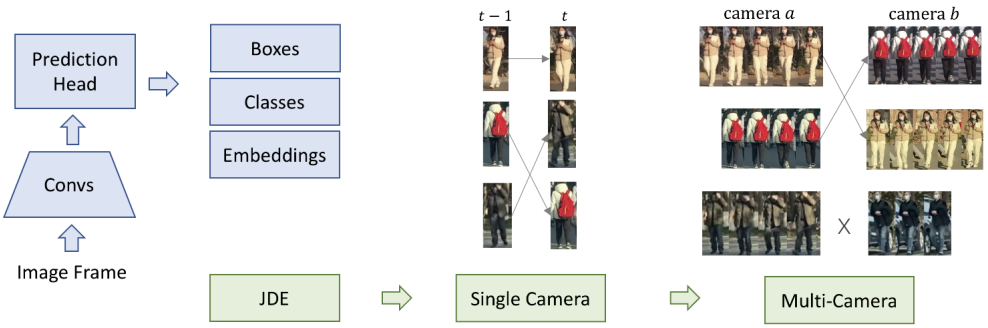

Figure 3. The workflow of MTMMC tracking, composed of three main parts: joint detection and embedding, single camera based tracking, and multi-camera based tracking.

4.1. Joint Detection and Embedding

The mainstream of most tracking algorithms lies in "tracking-by-detection", i.e., detection and embedding are usually treated as two independent steps. To improve the efficiency of tracking systems, in this paper we explore the joint detection and embedding procedures, which are expected to simultaneously give the location and appearance information of objects in a single forward pass. We have adapted the framework proposed in [37] to accomplish this task.

Feature Pyramid Network (FPN) [24] is employed to make the predictions from different scales. The objects in our dataset varies a lot regarding the scale. A better detection result is expected with FPN, a top-down architecture with skip connections for building high-level semantic feature maps at multiple scales. The prediction heads contain three components, i.e., one for box classification, one for box regression, and the other is for the embedding.

The detection branch evolves from the standard RPN proposed in [30]. Since the targets in our case is pedestrian, the aspect ratio is set as 1:3 due to common prior. The objective function of detection has two parts, cross-entropy loss for the foreground/background classification, and smooth-L1 loss for the bounding box regression.

To embed the feature for each target, we benefit from the metric learning and aim to learn a mapping from images to a compact Euclidean space where distances directly correspond to a measure of pedestrian similarity. Here, cross-entropy loss is applied for object identity classification in the same manner as [37, 5], which employs learnable class-wise weights as proxies of class instances rather than using the embeddings of instances directly. The purpose of this step is to minimize the distance between the same target in different frames, and maximizes the distance between different identities either in the same frame or different frames. The cross entropy loss for embedding branch is defined as Equation 1.

$$L = -log\frac{e^{f^{\perp}g^{+}}}{e^{f^{\perp}g^{+}} + \sum_i e^{f^{\perp}g^{-}}}, \tag{1}$$

where $f^{\perp}$ denotes the embedding of the anchor, while g^{+} and g^{-} represent the class-wise weight of the positive class (to which the anchor instance belongs) and weights of negative classes.

In summary, JDE will output predicted bounding boxes and appearance embeddings. The JDE module is treated as a multi-task learning problem. An automatic loss balancing scheme [18] is adopted by employing task-independent uncertainty. The learning objective function is written as,

$$L_{total} = \sum_{i}^{N} \sum_{j=\alpha,\beta,\gamma} \frac{1}{2}(\frac{1}{e^{s_j^i}} L_j^i + s_j^i), \tag{2}$$

where N is number of the prediction heads based on FPN layers. s_j^i are learnable parameters, representing the task-dependent uncertainty values. The joint objective depicts a weighted sum from every scale and every prediction component.

4.2. Single Camera based Online Association

In most cases, the objects move straightly on the road. The Kalman Filter (KF) is adopted to approximate the relative movements and infer the potential location of previous tracklets in the current frame. The wrong assignment can be excluded when an unconsistency is observed.

When the object detections and embedding features are available through JDE, a following tracking algorithm is needed to associate the same identities among consecutive frames. The problem is intuitive to be formulated using a bipartite graph for every two adjacent frames. We denote the detections as D^t and the corresponding feature embedding as f^t at frame t. The motion state in KF is denoted by m. The i-th object of $D_i^t \in D^t$ in frame t is expected to be matched with a live track $D_j^{t-1} \in D^{t-1}$ in frame $t-1$. The Hungarian method [20] finds a perfect matching such that

the cost meets some criterion. Considering both appearance information and motion information, we define the matching cost between the j-th track and the i-th detection as,

$$C = \lambda d_1(f_j, f_i^t) + (1-\lambda) d_2(m_j, m_i^t), \tag{3}$$

where $d_1(\cdot,\cdot)$ is a distance measure of Euclidean distance and $d_2(\cdot,\cdot)$ represents Mahalanobis distance. Every embedding and motion condition of the observations will be compared to a pool of previous existing tracks. The two matches of j-th track and i-th detection will be rejected if the cost value is over a threshold of φ_1.

The motion state of all matched tracklets are updated by KF. We assume f^t is the embeddings of the tracklet at frame t, and it can be updated following,

$$f^t = \eta f^{t-1} + (1-\eta)\tilde{f}, \tag{4}$$

where $\tilde{f}$ indicates the embedding of the assigned observation in frame t. η is a weight for balancing the historical and the current embeddings.

4.3. Multi-Camera based Tracking

Through the single camera based tracking, we obtain the tracklets in each camera, i.e., the same pedestrians in different frames have been linked together. The averaged embedding features for the same identity in a tracklet is calculated to generate a clip-level feature. Different tracklets will be described by the clip-level features. Due to the diversities of viewing perspectives and camera devices, a target's color-channel image intensities extracted from one camera are normally different from those of the other camera. But the JDE module can learn the deep features well based on large-scale training data. With the single camera based tracking result available, we model the multi-camera based tracking as a linear assignment problem. The fields of views of different cameras may be overlapping or non-overlapping. The tracklet in camera a may have a match in camera b or not, as is shown in Figure 3. The problem is solved by Jonker-Volgenant algorithm [16]. The cost function is defined as,

$$C = d(\frac{1}{T_1}\sum_{t=1}^{T_1} f_i^t, \frac{1}{T_2}\sum_{t=1}^{T_2} f_j^t), \tag{5}$$

where $d(\cdot,\cdot)$ is a distance measure of Euclidean distance between the i-th tracklet (T_1 frames) in the first camera and j-th tracklet (T_2 frames) in the other camera. It's important to choose a proper threshold to suppress the mismatches. If C is too large, the two tracklets will not be associated together. We use φ_2 to determine the association threshold.

5. Experiments

5.1. Experimental Settings

We have taken advantage of the pretrained model from [37] for the JDE task. The DarkNet-53 is implemented as the backbone network. Six datasets (ETH dataset, CityPersons, CalTech, MOT-16, CUHK-SYSU, and PRW dataset) regarding pedestrian detection, MOT, and person search have been put together to form a large-scale training set, resulting in better performance on detection accuracy. Rich images under different conditions of lighting, occlusion, weather, and viewing perspectives ensure a good generalization ability of the model.

Though we have performed Hungarian algorithm and KF for single camera based tracking, which are also the main approaches used in Wang et al. [37], we have several different implementations in the details. For example, an observation that is not assigned to any existing tracklets is initialized as a new tracklet, rather than waiting for accumulating two consecutive frames; when a track is re-found, we update the track's current location in time; and etc.

Here, we give the parameter settings. The N equals 3 in Equation 2, denoting three scales downsampled as, 1/8, 1/16, and 1/32 in FPN layers. As for updating the feature embedding of a tracklet, the momentum term of η is also set as 0.9, to fully consider the influence both from previous frames and the current frame. As for the cost functions in Section 4.2 and Section 4.3, we set the threshold of φ_1 as 0.8 and φ_2 as 1.0. The two parameters are empirically chosen according to the observations based on the data distribution. Besides, we define $\lambda = 0.98$ to balance the two parts of appearance and motion costs.

To make a comparison for different tracking method, we have also test other methods of Tracktor and DeepSort, implemented based on MMTracking [7].

5.2. Evaluation Metrics

We have used CLEAR-MOT metrics for evaluations [3]. In the multi-camera setting, the scores are computed based on the concatenated videos from all cameras which are in concern. The ground truth for the pedestrians are labeled with a consistent global ID across different cameras in each scene. We mainly use IDF1, IDP, IDR to evaluate the performance for multi-camera tracking. These ID measures describe how well the tracker recognizes who is where regardless of where or why mistakes occur. IDF1 measures the ratio of correctly identified detections over the average number of ground-truth and computed detections. A high IDF1 score is obtained when the correct multi-camera pedestrians are discovered, accurately tracked within each video, and labeled with a consistent ID across all videos in the dataset. IDP is the fraction of predicted detections that are correctly identified, while IDR is the fraction of true ob-

jects that are correctly detected. IDF1 is widely used as the principal measure for ranking MTMC trackers in the community [29]. To evaluate the performance for single-camera tracking, we also report MOTA [3], which accounts for all object configuration errors made by the tracker, false positives, misses, mismatches, over all frames.

5.3. Results

To implicitly show the performance of feature embedding module through JDE, we use t-Distributed Stochastic Neighbor Embedding (t-SNE) [14] to visualize the high-dimensional data by giving each datapoint a location in a two-dimensional map. The sequences in scene B and D are shown in Figure 4, from which we can see that the points with the same color tend to be grouped together. It means that the embedding module does give the discriminative feature for each unique identity.

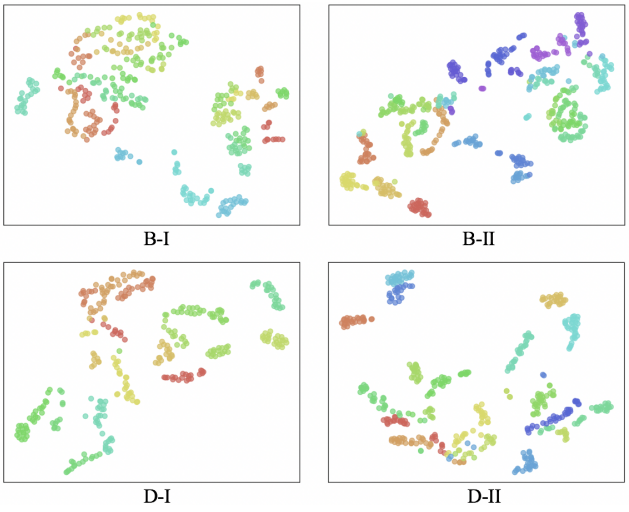

Figure 4. Visualization of feature embedding for different identities using t-SNE. The same color represents the same pedestrian in different frames.

We report the single camera based tracking result in Table 3. The overall IDF1, IDP, IDR, and MOTA are 66.5%, 73.1%, 60.8%, and 62.3%. We also observe some fluctuations among different sequences for these values. The results in D-I, D-II, A-II, B-I, and B-II are relatively more stable, either due to longer time duration or larger number of instances. The metrics are not good in A-I and F-III, these two sequences contain less tracks in relatively short time periods.

The multiple camera based tracking result is reported in Table 4. Scene B and D contains more instances than the others, the IDF1 values are 59.6% and 56.5%, respectively. These two scenes are also more crowded compared with others. The experimental results demonstrate the effectiveness of our proposed workflow for multiple pedestrian tracking across multiple moving cameras. We also show

Table 3. Results of single camera tracking

Sequence	IDF1↑	IDP↑	IDR↑	MOTA↑
A-I	38.1%	57.8%	28.1%	15.2%
A-II	62.0%	58.6%	65.9%	74.6%
B-I	72.0%	76.8%	67.9%	76.1%
B-II	60.8%	65.4%	56.9%	76.8%
C-I	69.5%	87.7%	57.6%	61.6%
C-II	65.1%	70.9%	60.1%	61.8%
D-I	65.3%	72.6%	59.4%	57.6%
D-II	56.5%	64.4%	50.3%	49.5%
E-I	79.1%	94.8%	67.8%	64.7%
E-II	85.0%	80.6%	89.9%	68.2%
E-III	91.2%	98.4%	85.0%	83.6%
F-I	70.2%	77.8%	64.0%	48.4%
F-II	74.1%	77.9%	63.8%	48.1%
F-III	28.7%	32.2%	25.9%	29.2%
OVERALL	66.5%	73.1%	60.8%	62.3%

Table 4. Results of multiple cameras tracking

Scene	IDF1↑	IDP↑	IDR↑
A	48.7%	51.6%	46.0%
B	59.6%	63.8%	55.9%
C	60.9%	67.2%	55.7%
D	56.5%	63.7%	50.8%
E	63.3%	69.8%	57.9%
F	48.8%	52.9%	43.2%
OVERALL	57.8%	63.6%	52.8%

the detected and tracked results in different scenes under different cameras in Figure 5. Specially, two cross-camera tracked pedestrians obtained through the tracking scheme described in Section IV are shown in Figure 6, where the bounding boxes are cropped from the videos and resized to the same size for better visualization. We can see that, though the color brightness can be different under different cameras, our proposed baseline can still accomplish the tracking across the moving cameras.

As is known to all, an effective module on single camera tracking usually leads to a better performance of multi-camera based tracking. Thus, it is necessary to exploit an approach which runs fast and tracks robustly. We compare the JDE module used in our proposed workflow with two other well-recognized works, i.e., DeepSort [38] and Tracktor [2]. Tracktor realizes multi-object tracking only with an object detector. It operates on the regression of the object detector, trying to align already existing track bounding boxes in frame $t-1$ to the object's new position at frame t. DeepSort learns a deep association metric and establish measurement-to-track associations using nearest neighbor

Figure 5. Exampled video frames from different sequences in Scene B, D, E, and F. The color brightness of devices is slightly different. The tracked boxes are also drawn in the images.

Figure 6. The same pedestrian in different cameras being assigned to the same identity through multi-camera based tracking methodology. Two examples from Scene B (two cameras) and E (three cameras) are given.

queries in visual appearance space. The results are reported in Table 5. Though Tracktor and DeepSort are also light modules which can run fast, but the performances are not as good as JDE employed in our framework. The embedding in the JDE trained by cross entropy loss are more discriminative for tracking different individuals. Besides, it is worthwhile to note that the proposed MTMMC tracking system provides a general solution for coordination among multiple moving cameras. It can adopt any newer and more powerful single camera tracking algorithm, whenever it is available.

Table 5. Comparisons of single-camera tracking methods

Sequence	Deepsort		Tracktor	
	IDF1	MOTA	IDF1	MOTA
A-I	5.6%	-1.8%	15.9%	4.7%
A-II	25.8%	0.3%	35.6%	-17.1%
B-I	22.2%	23.3%	51.1%	53.8%
B-II	21.6%	15.8%	53.0%	51.7%
C-I	37.4%	19.2%	46.4%	39.4%
C-II	22.3%	23.4%	35.5%	33.2%
D-I	38.2%	23.5%	54.9%	36.5%
D-II	25.1%	10.0%	51.9%	26.0%
E-I	56.9%	56.1%	71.9%	58.0%
E-II	94.5%	89.2%	95.8%	91.9%
E-III	64.0%	75.1%	88.0%	78.6%
F-I	20.4%	11.3%	59.5%	33.9%
F-II	43.7%	25.8%	67.5%	38.0%
F-III	13.2%	18.4%	50.7%	29.7%
OVERALL	33.2%	26.3%	55.5%	41.3%

6. Conclusion

In this paper, we explore the coordinated mining of different moving cameras for multi-pedestrian tracking on the road. Due to the lack of such publicly available dataset, we have collected a dataset, called "DHU-MTMMC", with multiple moving cameras enrolled in. The dataset contains several driving conditions for looking into different scenarios, both overlapping and non-overlapping camera views have been considered. We hope that the dataset will offer an opportunity for the researchers who are interested in this task. Besides, we also propose an MTMMC tracking system as a baseline to handle the multiple pedestrian tracking problem under different moving cameras. It consists of three stages: 1) joint detection and embedding, 2) single camera based tracking, and 3) multi-camera based tracking. The tracking is much challenging with the rapid changes of fields of views of the cameras, when the vehicles carrying the visual sensors are moving. Our proposed method achieves an overall IDF1 score of 57.8% on the proposed dataset. In the future, we will continuously keep enlarging the dataset and improving the tracking performance.

Acknowledgements

This work is supported by Shanghai Sailing Programs (21YF1401300, 20YF1401500) and the Fundamental Research Funds for the Central Universities (2232021D-25).

References

[1] J. Berclaz, F. Fleuret, E. Turetken, and P. Fua. Multiple object tracking using k-shortest paths optimization. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 33(9):1806–1819, 2011.

[2] Philipp Bergmann, Tim Meinhardt, and Laura Leal-Taixe. Tracking without bells and whistles. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 941–951, 2019.

[3] Keni Bernardin and Rainer Stiefelhagen. Evaluating multiple object tracking performance: the clear mot metrics. *EURASIP Journal on Image and Video Processing*, 2008:1–10, 2008.

[4] Alexey Bochkovskiy, Chien-Yao Wang, and Hong-Yuan Mark Liao. Yolov4: Optimal speed and accuracy of object detection. *arXiv preprint arXiv:2004.10934*, 2020.

[5] Jiarui Cai, Yizhou Wang, Haotian Zhang, Hung-Min Hsu, Chengqian Ma, and Jenq-Neng Hwang. Ia-mot: Instance-aware multi-object tracking with motion consistency. *arXiv preprint arXiv:2006.13458*, 2020.

[6] Wongun Choi. Near-online multi-target tracking with aggregated local flow descriptor. In *Proceedings of the IEEE international conference on computer vision*, pages 3029–3037, 2015.

[7] MMTracking Contributors. MMTracking: OpenMMLab video perception toolbox and benchmark. `https://github.com/open-mmlab/mmtracking`, 2020.

[8] Patrick Dendorfer, Aljosa Osep, Anton Milan, Konrad Schindler, Daniel Cremers, Ian Reid, Stefan Roth, and Laura Leal-Taixé. Motchallenge: A benchmark for single-camera multiple target tracking. *International Journal of Computer Vision*, pages 1–37, 2020.

[9] C. Ding, B. Song, A. Morye, J. A. Farrell, and A. K. Roy-Chowdhury. Collaborative sensing in a distributed ptz camera network. *IEEE Transactions on Image Processing*, 21(7):3282–3295, 2012.

[10] M. Farenzena, L. Bazzani, A. Perina, V. Murino, and M. Cristani. M.: Person re-identification by symmetry-driven accumulation of local features. In *In: IEEE Conf. Computer Vision and Pattern Recognition*, pages 2360–2367, 2010.

[11] Christoph Feichtenhofer, Axel Pinz, and Andrew Zisserman. Detect to track and track to detect. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 3038–3046, 2017.

[12] J. Ferryman and A. Ellis. Pets2010: Dataset and challenge. In *2010 7th IEEE International Conference on Advanced Video and Signal Based Surveillance*, pages 143–150, 2010.

[13] A Geiger, P Lenz, C Stiller, and R Urtasun. Vision meets robotics: The kitti dataset. *The International Journal of Robotics Research*, 32(11):1231–1237, 2013.

[14] Geoffrey Hinton and Sam T Roweis. Stochastic neighbor embedding. In *NIPS*, volume 15, pages 833–840. Citeseer, 2002.

[15] Hung-Min Hsu, Tsung-Wei Huang, Gaoang Wang, Jiarui Cai, Zhichao Lei, and Jenq-Neng Hwang. Multi-camera tracking of vehicles based on deep features re-id and trajectory-based camera link models. In *CVPR Workshops*, pages 416–424, 2019.

[16] Roy Jonker and Anton Volgenant. A shortest augmenting path algorithm for dense and sparse linear assignment problems. *Computing*, 38(4):325–340, 1987.

[17] Kai Kang, Hongsheng Li, Tong Xiao, Wanli Ouyang, Junjie Yan, Xihui Liu, and Xiaogang Wang. Object detection in videos with tubelet proposal networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 727–735, 2017.

[18] Alex Kendall, Yarin Gal, and Roberto Cipolla. Multi-task learning using uncertainty to weigh losses for scene geometry and semantics. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 7482–7491, 2018.

[19] T. Kokul, A. Ramanan, and U. A. J. Pinidiyaarachchi. Online multi-person tracking-by-detection method using acf and particle filter. In *2015 IEEE Seventh International Conference on Intelligent Computing and Information Systems (ICICIS)*, pages 529–536, 2015.

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

[21] Kuan-Hui Lee and Jenq-Neng Hwang. On-road pedestrian tracking across multiple driving recorders. *IEEE Transactions on Multimedia*, 17(9):1429–1438, 2015.

[22] Kuan-Hui Lee, Jenq-Neng Hwang, Greg Okapal, and James Pitton. Driving recorder based on-road pedestrian tracking using visual slam and constrained multiple-kernel. In *17th International IEEE Conference on Intelligent Transportation Systems (ITSC)*, pages 2629–2635. IEEE, 2014.

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

[24] 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.

[25] Roberto Marroquin, Julien Dubois, and Christophe Nicolle. Wisenet: An indoor multi-camera multi-space dataset with contextual information and annotations for people detection and tracking. *Data in brief*, 27:104654, 2019.

[26] Alexis Mignon and Frédéric Jurie. Pcca: A new approach for distance learning from sparse pairwise constraints. In *2012 IEEE conference on computer vision and pattern recognition*, pages 2666–2672. IEEE, 2012.

[27] Anton Milan, Laura Leal-Taixé, Ian Reid, Stefan Roth, and Konrad Schindler. Mot16: A benchmark for multi-object tracking. *arXiv preprint arXiv:1603.00831*, 2016.

[28] Anton Milan, S Hamid Rezatofighi, Anthony Dick, Ian Reid, and Konrad Schindler. Online multi-target tracking using recurrent neural networks. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 31, 2017.

[29] Milind Naphade, Shuo Wang, David C Anastasiu, Zheng Tang, Ming-Ching Chang, Xiaodong Yang, Liang Zheng,

Anuj Sharma, Rama Chellappa, and Pranamesh Chakraborty. The 4th ai city challenge. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*, pages 626–627, 2020.

[30] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. *arXiv preprint arXiv:1506.01497*, 2015.

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

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

[33] Olly Styles, Tanaya Guha, Victor Sanchez, and Alex Kot. Multi-camera trajectory forecasting: Pedestrian trajectory prediction in a network of cameras. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*, pages 1016–1017, 2020.

[34] Siyu Tang, Mykhaylo Andriluka, Bjoern Andres, and Bernt Schiele. Multiple people tracking by lifted multicut and person re-identification. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3539–3548, 2017.

[35] 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. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8797–8806, 2019.

[36] Gaoang Wang, Yizhou Wang, Haotian Zhang, Renshu Gu, and Jenq-Neng Hwang. Exploit the connectivity: Multi-object tracking with trackletnet. In *Proceedings of the 27th ACM International Conference on Multimedia*, pages 482–490, 2019.

[37] Zhongdao Wang, Liang Zheng, Yixuan Liu, and Shengjin Wang. Towards real-time multi-object tracking. *arXiv preprint arXiv:1909.12605*, 2(3):4, 2019.

[38] Nicolai Wojke, Alex Bewley, and Dietrich Paulus. Simple online and realtime tracking with a deep association metric. In *2017 IEEE international conference on image processing (ICIP)*, pages 3645–3649. IEEE, 2017.

[39] Ekim Yurtsever, Jacob Lambert, Alexander Carballo, and Kazuya Takeda. A survey of autonomous driving: Common practices and emerging technologies. *IEEE Access*, 8:58443–58469, 2020.

[40] Yifu Zhang, Chunyu Wang, Xinggang Wang, Wenjun Zeng, and Wenyu Liu. Fairmot: On the fairness of detection and re-identification in multiple object tracking. *arXiv preprint arXiv:2004.01888*, 2020.

[41] Wei-Shi Zheng, Shaogang Gong, and Tao Xiang. Towards open-world person re-identification by one-shot group-based verification. *IEEE transactions on pattern analysis and machine intelligence*, 38(3):591–606, 2015.

[42] Pengfei Zhu, Longyin Wen, Dawei Du, Xiao Bian, Qinghua Hu, and Haibin Ling. Vision meets drones: Past, present and future. *arXiv preprint arXiv:2001.06303*, 2020.