MMPTRACK: Large-scale Densely Annotated Multi-camera Multiple People Tracking Benchmark

Xiaotian Han Quanzeng You Chunyu Wang Zhizheng Zhang Peng Chu
Houdong Hu Jiang Wang Zicheng Liu
Microsoft
{xiaothan, quyou, chnuwa, zhizzhang, pengchu, houhu, jiangwang, zliu}@microsoft.com

Abstract

Multi-camera tracking systems are gaining popularity in applications that demand high-quality tracking results, such as frictionless checkout. In cluttered and crowded environments, monocular multi-object tracking (MOT) systems often fail due to occlusions. Multiple highly overlapped cameras are capable of recovering partial 3D information. When used properly, 3D data can significantly alleviate the occlusion issue. However, training a multi-camera tracker demands a large-scale multi-camera tracking dataset with diverse camera settings and backgrounds. These requirements make the collection of multi-camera tracking dataset challenging and expensive. The cost of creating such a dataset has limited the availability and scale of datasets in this domain. Instead, we appeal to an auto-annotation system to reduce the cost, which uses overlapped and calibrated depth and RGB cameras to build a 3D tracker and automatically generates the 3D tracking results. The results are manually checked and corrected to ensure the label quality, which is much cheaper than solely manual annotation. Next, the 3D tracking results are projected to each calibrated RGB camera view to create 2D tracking results. In this way, we collect and annotate a large-scale densely labeled multi-camera tracking dataset from five different environments. We have conducted extensive experiments using two real-time multi-camera trackers and a person re-identification (ReID) model under different settings. This dataset provides a reliable benchmark for multi-camera, multi-object tracking systems in cluttered and crowded environments. We expect this benchmark to encourage more research attempts in this domain. Our dataset will be publicly released upon the acceptance of this work.

1. Introduction

Multiple object tracking (MOT) [7, 31] is one of the fundamental research problems in computer vision. As more efficient and powerful deep neural networks are continuously being developed, the accuracy of MOT systems has been substantially improved in recent years. However, monocular MOT systems still make tracking errors in cluttered and crowded environments, where occlusions of the tracked objects often occur. Thus, such systems may be inadequate for applications that require highly accurate and consistent tracking results, such as frictionless checkout in retail stores or autonomous driving.

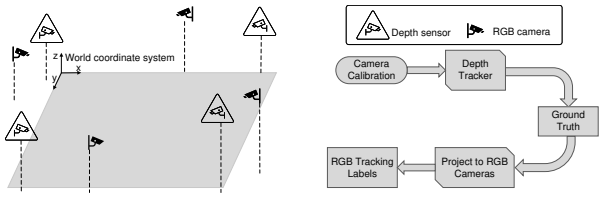

Figure 1: The auto-annotation system consists of multiple calibrated depth sensors and RGB cameras to build a 3D tracker, which generates pseudo ground-truth 3D tracking results. The tracking results are manually corrected and projected to each RGB camera view as tracking labels.

Recently, multi-camera systems have been widely deployed in these applications [1]. The overlapped and calibrated cameras can considerably remedy the occlusion issue. As a result, multi-camera tracking systems have achieved much higher accuracy than single-camera tracking systems [48]. However, only a few small multi-camera datasets are publicly available due to data collection and annotation challenges. The lack of high-quality training and evaluation data makes it difficult to further improve current multi-camera tracking systems.

In this paper, we collect and annotate a large-scale multi-camera multi-object tracking dataset, which consists of full-body bounding boxes and consistent tracking IDs across all RGB camera views as well as the shared top-down view. The labels are annotated by an auto-annotation system, which utilizes depth sensors to construct a high-performance 3D tracker. Figure 1 illustrates the overview

of our system. It consists of multiple calibrated depth sensors and RGB cameras. We follow the design in [44], where the 3D tracker works on the projected top-down view of the 3D space constructed from depth sensors. We train a person detector on the projected top-down view and follow the tracking-by-detection framework [3] to build the 3D tracker. The 3D tracker produces consistent and accurate tracking results on the projected top-down view. We further ask human annotators to correct the 3D tracking errors, such as ID switches and false-positive tracks. The corrected per-frame 3D tracking results are projected to all synchronized RGB streams using the camera parameters. Our experiments show that the auto-annotation system can produce high-quality tracking annotations (100% IDF1 and 99.9% MOTA) and reduce the labor cost to $1/800$ of the traditional annotation methods.

We set up five challenging environments in our lab. With the help of the auto-annotation system, we construct the largest multi-camera multiple people tracking dataset so far. The dataset is densely annotated, *e.g.*, per-frame full-body bounding boxes and person identities are available. We evaluate two state-of-the-art real-time multi-camera person trackers [45, 48] and a person re-identification (ReID) model [50] on our dataset under various settings. Our experiments demonstrate that the detectors, trackers, and ReID models trained on publicly available datasets, such as MS-COCO [29] or MSMT [40], do not perform well in these challenging environments because of viewpoint differences and domain gaps. However, adapting the models using the training split of the data can significantly improve the accuracy. We expect the availability of such large-scale multi-camera multiple people tracking dataset will encourage more participants in this research topic. This dataset is also valuable for the evaluation of other tasks, such as multi-view people detection [20, 28] and monocular multiple people tracking [7]. To summarize, our contributions are as follows:

- We construct the largest densely annotated multi-camera multiple people tracking dataset to encourage more research on this topic.
- We propose an auto-annotation system to produce high-quality tracking labels for multi-camera environments in a fast and cost-efficient way.
- We conduct extensive experiments to reveal the challenges and characteristics of our dataset.

2. Related work

Approaches Multi-camera multi-object tracking has been extensively studied in computer vision community. Previously, different graph-based approaches have been proposed to solve the data associations across different frames and cameras [5, 46, 18, 35, 38, 11, 41, 12, 17]. Recent approaches [33, 43, 37, 22] attempt to apply deep ReID features for the data association. Extra efforts are needed to handle cross-camera appearance changes [23, 21]. These methods can be applied to environments with non-overlapping cameras, but they cannot explicitly utilize the camera parameters for cross-camera association and 3D space localization.

Other approaches adopt camera calibration for tracklets merging and cross-camera association. Probabilistic occupancy map (POM) [15] is one of the early representative studies. POM provides a robust estimation of the ground-plane occupancy, which is the key to building a high-performance tracker in crowded environments. Also, homography [13] is employed to merge head segments from all camera views to build a head tracker. Later, deep occlusion [4] extends this idea by utilizing Convolutional Neural Network (CNN) and Conditional Random Field (CRF) to reason the occlusions.

Recently, 3D pose estimation and 3D person detection methods are utilized for multi-camera people tracking [48]. 3D pose can be estimated [36, 10] by merging 2D skeleton estimations from multiple 2D camera views, using a 3D regression network or graph matching. Meanwhile, multi-view person detection approaches [20, 28, 34, 19] utilize camera calibration to merge multiple 2D detections or features to generate more reliable 3D person detection results. These approaches heavily depend on the quality of the 2D person detection or 2D pose estimation. These 3D poses and detections can be utilized for 3D trackers.

Datasets Several multi-camera tracking datasets with highly overlapping cameras, have been adopted in multi-target multi-camera tracking research. Among them, PETS2009 [14], Laboratory [15], Terrace [15], Passageway [15], USC Campus [25] and CamNet [47] have been collected by low-resolution cameras and only have a small number of frames and person identities (IDs). EPFL-RLC [9], CAMPUS [42] and SALSA [2] are released more recently. However, EPFL-RLC only has 300 annotated frames, and CAMPUS comes without 3D ground truth. WILDTRACK dataset [8] consists of high-quality annota-

Dataset	# of Envs	Cameras	FPS	Overlap	Calib	GT (frames)	Length (minutes)
USC Campus[25]	1	3	30	No	No	135,000	25
CamNet[47]	6	8	25	Yes	No	360,000	30
DukeMTMC[32]	1	8	60	No	Yes	2,448,000	85
SALSA[2]	1	4	15	Yes	Yes	$\sim$1,200	60
WILDTRACK[8]	1	7	60	Yes	Yes	$\sim 7\times 9{,}518$	60
MMPTRACK (ours)	5	23	15	Yes	Yes	$\sim$2,979,900	576

Table 1: Representative multi-camera person tracking datasets. FPS stands for frame per second.

tions of both camera-view and 3D ground truth, as well as more person identities. However, the annotations are sparse and limited to 400 frames. DukeMTMC [32] is released with over 2 million frames and more than 2700 identities. However, there are almost no overlaps among different cameras. Table 1 compares our dataset (MMPTRACK) with several existing datasets. MMPTRACK is captured with a large number of calibrated overlapped cameras in indoor environments, which aligns better with the applications such as frictionless checkout. MMPTRACK is much larger than the existing data both in terms of the video length and the number of annotated frames. The videos are labelled frame-wise with full-body bounding boxes and consistent person identities cross all cameras.

3. Dataset collection

3.1. Dataset statistics

The statistics of the collected dataset are summarized in Table 2. Our dataset is recorded with 15 frames per second (FPS) in five diverse and challenging environments. Overall, we collect about 9.6 hours of videos, with over half a million frame-wise annotations for each camera view. This is by far the largest publicly available multi-camera multiple people tracking (MMPTRACK) dataset.

Envs	Retail	Lobby	Industry	Cafe	Office	Total
# of cameras	6	4	4	4	5	23
Train (min)	84	65	52	14	46	261
Validation (min)	43	32	31	28	19	153
Test (min)	45	32	32	31	22	162
Total (min)	172	129	115	73	87	576

Table 2: Statistics of Multi-camera Multiple People Tracking (MMPTRACK) dataset.

Figure 2 shows examples of tracking labels of each camera view from two different environments. Although both environments are crowded and cluttered, our ground truth exhibits high-quality full-body bounding boxes and consistent person IDs across all camera views.

3.2. Environment setup

We set up 5 different environments in our lab, *i.e.*, *Cafe Shop*, *Industry*, *Lobby*, *Office* and *Retail*. We install Azure Kinect cameras in every environment with fixed positions and view angles. Figure 3 shows the field-of-view overlaps among different cameras on the ground plane within each environment. Azure Kinects can record RGB and depth streams simultaneously. Their RGB streams are used as the default RGB cameras for our dataset (see Figure 1)[1]. Depth and RGB streams are recorded synchronized within and across Azure Kinects.

[1]Other RGB cameras can also be used for data collection as long as calibrated with existing Azure Kinect cameras.

3.3. Camera calibration

Intrinsic parameters We obtain Azure Kinect intrinsic parameters directly from its SDK. We denote intrinsic parameters as I.

Extrinsic parameters In our settings, one camera has overlapping field of view with at least another camera. We use ArUco markers from OpenCV library[2] as reference points in the world coordinate system. We build a connected bipartite graph, where cameras and ArUco markers are vertices. If ArUco marker m_i is within the view of camera c_j, we will add an edge e_{ij} between them. Figure 4 shows an example of the connected bipartite graph for calibration of the *Industry* environment. Let $P = \bigcup P_i$ be the set of detected corner points of all markers (P_i is the corners from i-th marker). Then, the set of extrinsic parameters E are obtained by optimizing

$$E^* = \arg\max_{E,M} \sum_{i=1}^{|P|} \sum_{c=1}^{C} \mathbb{1}_i^c \|p_i^c - I^c * E^c * m_i\|^2, \quad (1)$$

where $\|\cdot\|$ denotes Euclidean distance, $\mathbb{1}_i^c$ is an indicator function, whose value equals to 1 only if point p_i is visible in camera view c, $M = \{m_i, i = 1, \cdots, |P|\}$ denotes the markers' corner points in world coordinate system. The graph optimization approach proposed in [26] is implemented to solve Eq. (1).

3.4. Dataset collection

Our dataset is recorded in four half-day sessions. In each session, we hired seven different subjects to participate. Each subject can act improvisationally as long as their action fits the environment setting. For instance, in *Retail* environment, they are free to perform any shopping behaviors, *e.g.*, pushing shopping carts, holding baskets, and standing in a queue for checkout; while in Cafe environment, they can sit together, drinking , chatting, *etc*. Following such instructions, the collected dataset covers a wide variety of people's behaviors. In total, we have 28 subjects, with different ages, genders, and ethnicities, which provides enough fairness and diversity to our dataset.

3.5. 3D auto-annotation system

Our 3D auto-annotation system utilizes depth streams to perform high quality 3D person tracking. The workflow of the system is described in Algorithm 1. We build our 3D tracker using data released in [44], which does not have any overlap with the current dataset in terms of environment or

[2]https://docs.opencv.org/4.x/d5/dae/tutorial_aruco_detection.html

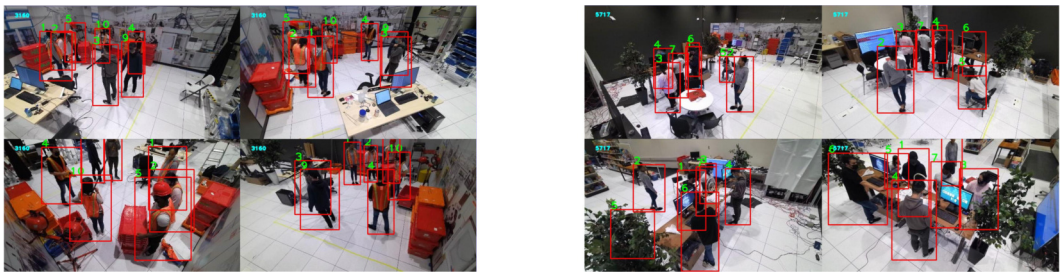

Figure 2: Examples of the images and tracking annotations of our dataset. **Left** and **Right** images are from *Industry* and *Lobby* environments, respectively.

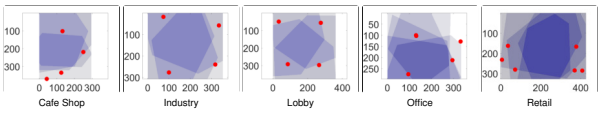

Figure 3: Overlaps among camera views of each environment on the ground plane. Each red dot represents the location of a camera. The X and Y axes represent the size of each environment in terms of pixels (each pixel unit is $20mm$).

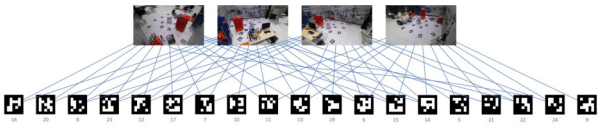

Figure 4: Bipartite graph from cameras and ArUco markers when calibrating cameras in *Industry* environment.

subject. The top-down view image is constructed from the merged 3D point cloud. This design avoids environment-dependent factors, such as lighting, camera angles, *etc*. Therefore, the 3D tracker can be easily generalized and applied to different environments.

Algorithm 1 Workflow of RGBD Auto-annotation System

Input: Synchronized RGB and depth steams and camera parameters C
Output: Full-body person bounding boxes and IDs in each camera view

procedure AUTO-ANNOTATION
  $B \leftarrow list()$ ▷ Person bounding boxes
  while `All Streams not end` **do**
    $R \leftarrow set()$ ▷ Synchronized RGB images
    $D \leftarrow set()$ ▷ Synchronized Depth images
    for `stream in Streams` **do**
      $r, d \leftarrow$ stream.read()
      add(r, R)
      add(d, D)
    end for
    $P \leftarrow$ **PointCloudGen**(R, D, C)
    $T_d \leftarrow$ **TopdownViewGen**(P) ▷ Top-down view of the scene
    $B \leftarrow$ **PersonDetector**(T_d)
    $T_r \leftarrow$ **3DTracker**(B, P) ▷ 3D Tracklets
    $B_c \leftarrow$ **Projection**(T_r, C) ▷ Camera-view bounding boxes
    append(B_c, B)
  end while
  return B
end procedure

Point cloud reconstruction We reconstruct the point cloud of the whole scene from calibrated and synchronized depth cameras. Given the intrinsic parameters I and extrinsic parameters E, the point cloud $\mathbb{P}$ is calculated as follows:

$$\mathbb{P} = \bigcup_{c=1}^{C} \bigcup_{i} \bigcup_{j} (E^c)^{-1} * (I^c)^{-1} * \left[i, j, d^c_{i,j}\right]^T \quad (2)$$

where i and j index over all valid locations and d^c_{ij} denotes camera c's depth measurement at location (i, j).

Top-down view projection We discretize the point cloud $\mathbb{P}$ into a binary voxel set $\mathbb{V}$. Each voxel $\mathbb{V}_{i,j,k}$ covers a cube with a volume of $20mm \times 20mm \times 20mm$. $\mathbb{V}_{i,j,k} = 1$ if and only if there exists at least one point $\mathbb{P}_{i',j',k'}$, such that it locates inside the cube covered by $\mathbb{V}_{i,j,k}$.

We set the world-coordinate system's X and Y axes parallel to the ground plane and the Z axis vertical to the ground. The top-down view image T_d can be obtained by projecting $\mathbb{V}$ onto the X-Y ground plane. More specifically, its value at position (m, n) is computed as:

$$T_d(m, n) = \underset{z, V(m,n,z)=1}{\arg\max} \ \mathbb{V}_{m,n,z}, \quad (3)$$

which can be perceived as the *height* of filled voxels within each cube $V_{m,n,(\cdot)}$.

Top-down view person detection We design a simple two-stage top-down person detector. Recall that pixel values of T_d (Eq. (3)) represent the height of each location. Therefore, in the proposal generation stage, we extract all local maxima from the top-down view image T_d. For each candidate at (i, j), we crop a 50×50 square region centered around it. The cropped image region is fed into a Convolutional Neural Network (a variant of ResNet-18), which serves as a person classifier in the second stage.

3D tracker The inputs of our tracker are top-down view detection boxes with corresponding detection scores and cropped point clouds. At the beginning frame, we initialize a tracklet when the detection score of a bounding box is above a threshold. For the following frames, we construct the cost matrix based on spatial and appearance (color histogram) distance between each tracklet and the detected bounding boxes. Association results are obtained by

Envs	IDF1↑	MOTA↑	FP↓	FN↓	IDs↓
Cafe	100	100	0	0	0
Industry	100	100	0	0	0
Lobby	100	100	0	0	0
Office	100	100	0	0	0
Retail	100	99.9	0	4	0

Table 3: Performance of our 3D tracker on one testing sequence.

employing Hungarian Matching algorithm. For each unmatched detection bounding box, we generate a new candidate tracklet.

Camera-view projection The height h of each tracked person can be estimated from the local maxima of its top-down bounding box. We fit a cube with a size of $100cm \times 100cm \times h$ centered at each top-down tracked person. The 3D bounding box (cube) is projected to each camera view. The 2D full-body bounding box in each view is the tightest rectangle that encloses the projected 3D bounding box in this view. In this way, we propagate the tracking results from 3D space to all RGB cameras.

3.6. Annotation and quality control

The 3D tracker may still introduce errors occasionally. We manually fix all tracking errors before propagating the results to each RGB camera view. The most common errors in our 3D tracker are tracklet ID switch and false-positive person detection. We request annotators to correct ID switches and remove false-positive tracklets from 3D tracking results. Notice this process is relatively cost-efficient because no bounding box labeling is required, and all the corrections are performed at the tracklet level.

Based on our experiments, each annotator can label around 600 frames (including boxes and IDs) per day for videos with around 5 to 6 persons inside. Manually labeling all the videos in our dataset costs 414 labeller days if annotating every 10 frames and interpolating the tracking labels to remaining frames. In comparison, we only need one labeller work less than 5 hours to correct all the errors of our 3D tracker.

To test the quality of the auto-generated ground truth, we sample $1,000$ continuous frames from each environment and manually label each frame. Table 3 summarizes the evaluation results of our corrected auto-generated ground truth against manual labels. Only four human-labeled boxes mismatch the ground truth of our dataset, which is tolerable given that humans can also make errors.

4. Benchmarks

In this section, we discuss the evaluation metrics, evaluated approaches and experimental results on both *tracking* and *ReID* tasks.

4.1. Evaluation metrics

For tracking task evaluation, we follow the widely adopted MOT metrics [7]. We report the false positive (**FP**) and false negative (**FN**) detections, which are also considered in multiple object tracking accuracy (**MOTA**). MOTA further deals with identity switches (**IDs**) and is extensively used in benchmarking different trackers. Besides, we also report **IDF1**, which measures the ID consistency between the predicted trajectories and the ground truth using ID precision and ID recall. IDF1 can assess the trackers' ability on tracklet association. We report all performance metrics on the top-down view for multi-camera tracking models. We follow the settings in [8], where a radius of one meter is used as the distance threshold when matching detections and ground truth.

For the ReID task, we adopt the widely used Rank-1 accuracy (**R-1**) and mean Average Precision (**mAP**) [50] to compare the model's performance under different settings.

4.2. Baselines trackers

We evaluate two state-of-the-art online real-time multi-camera trackers on our datasets. We focus on evaluating online real-time trackers because they can better reflect the core detection and tracking performance, and we can better observe the challenges in our dataset in these evaluations.

End-to-end deep multi-camera tracker (DMCT) In this baseline, we employ the end-to-end approach (DMCT) proposed in [45]. This approach estimates the ground point heatmap of each candidate at each camera view, projects the ground point heatmaps from all camera views to the ground plane, and fuses all the heatmaps into a ground-plane heatmap. Similar to [45], we train a variant of CornerNet [27] with pixel-wise Focal Loss [27] as our ground-point estimation model. The tracker works on the fused ground-plane heatmap.

Given the fused ground-plane heatmap H, two different approaches are utilized to obtain top-down person detections. The first approach is rule-based. It directly applies Gaussian blur to H and extracts local maxima as person detections for tracking. In this approach, the heatmaps $\{H_1, H_2, \cdots, H_C\}$ from C camera views are projected to the ground plane using homographies between the ground plane and all camera views. For each location in fused top-down heatmap H, its value is the maximum over all camera view's projected heatmap.

The second approach trains a YOLOV5 [24] detector as top-down person detector. The second approach is more expensive than the first approach, but it is much cheaper than the sequence-based deep glimpse network in [45]. In this approach, we first find the local maxima at each camera-view heatmap H_i as candidate points. These points are projected to the ground plane, and each point generates a Gaussian distribution around it to reduce noise. We keep

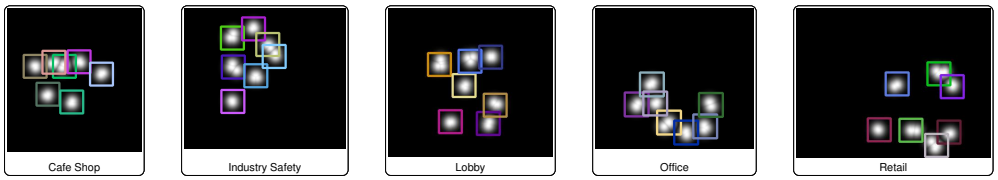

Figure 5: Examples of the fused ground-plane images as inputs to person detector. The bounding boxes are the ground truth.

the maximum value from all Gaussian distributions for each location in the projected top-down image. Figure 5 shows examples of fused ground-plane heatmaps from the five environments. Although the five environments are configured with different RGB camera settings and backgrounds, their top-down heatmaps look similar. Our experimental results show that the top-down detector can easily generalize across different environments.

We also label another external dataset using images from OpenImage[3]. Specifically, we sample a subset of OpenImage containing persons (about $600,000$ images), then manually label the ground point of each person in these images. In our experiments, we also study the impact of adding external data when training our person ground point detector on tracking performance.

The variants of DMCT approach include: **DMCT** trains the ground-point estimation model on the training split of MMPTRACK and the rule-based approach for top-down person detection; **DMCT-TD** uses the same ground-point estimation model with DMCT and the deep learning-based top-down person detector; **DMCT-Ext** uses the same rule-based top-down person detection with DMCT, and it trains ground-point estimation model with both training split of MMPTRACK and the extra manually labeled OpenImage dataset; **DMCT-Ext-TD** uses the same ground-point estimation model with DMCT-Ext, and the deep learning-based top-down person detector.

Tracking by 3D skeletons (VoxelTrack) This baseline performs tracking with estimated 3D body joints, which contain more spatial information than the single ground point. It is built on top of a state-of-the-art 3D pose estimation method VoxelPose [36]. It requires neither camera-view 2D pose estimation nor cross-camera pose association as in previous works, which is error-prone. Instead, all hard decisions are postponed and made in the 3D space after fusing 2D visual features from all views, which effectively avoids error accumulation. In addition, the fused representation is robust to occlusion. A joint occluded in one camera view may be visible in other camera views.

We follow a standard pipeline [49] for tracking the 3D poses. We initialize every estimated 3D pose as a tracklet. For the following frames, we use the Hungarian algorithm to assign the 3D poses to the existing tracklets, where the matching cost is the sum of the Euclidean distance for all the 3D joints. We reject the assignment if the spatial distance between the tracklet and the 3D pose is too large. An unmatched 3D pose will be assigned as a new tracklet. An existing tracklet will be removed if it is not matched to any 3D poses for more than 30 frames.

Following the settings of [36], the 2D heatmap estimation model is trained on the COCO dataset. Since MMPTRACK lacks 3D pose labels, we fine-tune the 3D model using synthetic data instead of real data. Calibration parameters of MMPTRACK are employed to generate pseudo-3D human poses.

4.3. Baseline ReID models

We evaluate a person re-identification (ReID) model proposed in FastReID [16] on the MMPTRACK dataset to test model robustness. We study the challenges of learning discriminative ReID features in a cluttered and crowded environment under multiple cameras. Our baseline model is built upon a commonly used baseline model [30]. We further incorporate Non-local block [39], GeM pooling [6] and a series of training strategies (see details in [16]).

We uniformly sample the MMPTRACK dataset every 32 frames. For testing, we divide each sampled sequence into two halves. We use the cropped persons in the first half as the query set and those in the second half as the gallery set. Although there are only a small number of person identities in this dataset, the diverse camera angles sampled cluttered background and various person actions make ReID a challenging task on our dataset.

We evaluate three training configurations of the above model on the testing split of MMPTRACK. Specifically, for the **Generalization** setting, we directly evaluate the model trained with the person ReID dataset MSMT [40]. For the **Adaptation** setting, we perform supervised fine-tuning of the previous model with cropped persons on the training split of MMPTRACK. For the **Supervised** setting, we train the person ReID model from scratch using only the cropped persons in the training split of MMPTRACK.

4.4. Benchmark results and discussions

4.4.1 Tracking performance on MMPTRACK

We evaluate the two real-time baseline trackers on the collected MMPTRACK dataset.

[3] https://storage.googleapis.com/openimages/web/index.html

Method	IDF1↑	MOTA↑	FP↓	FN↓	IDs↓
VoxelTrack	55.2	79.6	43,776	110,239	4,365
DMCT	60.2	91.5	34,450	41,920	2,158
DMCT-TD	74.8	93.6	15,080	42,854	620
DMCT-Ext	61.1	92.5	30,789	36,631	1,953
DMCT-Ext-TD	77.5	94.8	19,235	28,505	567

Table 4: Tracking performance on validation split.

Method	IDF1↑	MOTA↑	FP↓	FN↓	IDs↓
VoxelTrack	50.8	76.8	49,881	142,380	4,922
DMCT	56.0	88.8	39,715	52,559	2,677
DMCT-TD	68.1	93.2	16,023	40,606	935
DMCT-Ext	56.6	89.0	42,413	48,039	3,013
DMCT-Ext-TD	74.1	94.6	7,005	38,296	641

Table 5: Tracking performance on testing split.

Table 4 and Table 5 include the results of different baseline trackers on the validation and testing splits of our dataset respectively. The results suggest that the top-down person detector trained with a deep learning model can significantly boost the performance of baseline DMCT, especially for **IDF1** and **IDs**. With an extra of $600,000$ images from OpenImage, **DMCT-Ext** shows slight improvements over **DMCT** in terms of **IDF1** and **MOTA**. **DMCT-Ext-TD** improves **IDF1** by 2.7 over **DMCT-TD**. However, the **MOTA** only increases 1.1. Compared with **VoxelTrack**, which is only virtually fine-tuned on MMPTRACK, all variants of **DMCT** perform better. We believe the performance gap is due to the large domain differences of our dataset and other public datasets such as MS-COCO, on which the **VoxelTrack** is trained. Since we can easily generate a large-scale multi-camera multi-object tracking dataset for an environment using our auto-annotation system, we can train a model that adapts to a given environment with improved accuracy. However, the accuracy of the baseline methods is still not high enough to meet the requirements of demanding applications, particularly for IDF1. Further research is needed in this domain.

Ablation studies We study the impact of different training splits on our baselines' performance. Since **VoxelTrack** can only be virtually fine-tuned on our dataset, we only cover the ablation study results of different variants of **DMCT**. We attempt to study the impact of environment-specific data. In particular, we train the ground point estimation models and the top-down detectors with and without each environment-specific data. Then, we report the performance in each environment individually.

Table 6 shows the tracking evaluation metrics without the top-down detector. Generally, without each environment's data, the tracking performance drops significantly. This is particularly true in terms of the IDF1 metric. Also, external training data can improve the tracker's performance for most environments when environment-specific data is absent. However, with environment-specific data, external data leads to a limited performance gain.

Table 7 further studies DMCT's performance when equipped with a deep learning-based top-down detector. A tracker with a deep learning-based top-down detector has better generability. Without external data, models trained with environment-specific data show better performance in *Industry* and *Retail* in terms of **IDF1**. However, models trained without environment-specific data even show better **IDF1** in *Cafe*, *Lobby* and *Office*. Also, when extra OpenImage data is utilized to train a ground point estimation model, the performance gain is limited, and in some environments, even worse than the results without the external data. It is generally believed that a pre-trained model on external data may provide good initialization when training the deep model. However, the domain gap between MMPTRACK and OpenImage makes the pre-training step insignificant. Instead, the large-scale in-domain MMPTRACK dataset can train a model with better performance. Meanwhile, compared with Table 6, the results in Table 7 also suggests that the deep-learning-based top-down detector reduces the performance gap caused by external ground-point data.

4.4.2 ReID performance on MMPTRACK

We report the results of the ReID model with the three different settings discussed in Section 4.3. The MSMT ReID dataset, which is employed to pretrain our **Generalization** and **Adaptation** model, consists of more than $4,000$ indoor and outdoor person identities. The evaluation results are summarized in Table 8. Even though our training dataset consists of only 14 different person identities, training from scratch still outperforms the **Generalization** model. Notice that the person identities do not overlap in training and testing split. This shows that our large-scale dataset can help learn discriminative ReID features. Also, the fine-tuned model (**Adaptation**) is superior to the model trained from scratch (**Supervised**). Meanwhile, the performance of all models varies across different environments. All models perform poorly in *Retail* environment due to its cluttered background. In general, the experiment shows that Re-ID is very challenging in cluttered and crowded environments in multi-camera settings. Our large-scale dataset can help learn a more discriminative Re-ID feature that is adapted to a given environment. However, the performance is still far from satisfactory in a challenging environment. We believe that more identities are needed to learn more discriminative Re-ID features in these challenging environments.

5. Conclusion

In deep learning, high-quality labelled data is key to many tasks. This is particularly the case for multi-camera multiple people tracking, where trackers' performances are profoundly impacted by environment settings. In this work,

Method	w/Env Data	Env	Without external data					With external data				
			IDF1↑	MOTA↑	FP↓	FN↓	IDs↓	IDF1↑	MOTA↑	FP↓	FN↓	IDs↓
DMCT	✗	Cafe	39.4	87.7	12,012	7,589	1,158	56.9	91.8	6,792	6,382	701
DMCT	✓	Cafe	64.2	95.9	2,691	4,063	162	61.3	96.0	2,104	4,409	297
DMCT	✗	Industry	34.2	78.4	18,486	20,548	1,637	42.7	82.7	11,700	19,858	962
DMCT	✓	Industry	61.7	90.5	9,107	8,431	306	64.5	91.2	9,188	7,155	233
DMCT	✗	Lobby	47.1	86.4	13,774	12,223	1,136	50.6	89.6	12,574	7,425	720
DMCT	✓	Lobby	69.4	94.5	3,343	7,361	318	69.2	95.1	2,445	7,007	247
DMCT	✗	Office	50.0	89.0	3,287	8,577	591	40.0	85.8	2,849	12,569	735
DMCT	✓	Office	68.0	93.7	1,514	5,625	67	66.8	93.9	1,454	5,304	127
DMCT	✗	Retail	27.7	60.7	79,443	15,788	3,170	30.4	70.7	50,713	20,060	2,667
DMCT	✓	Retail	45.7	85.8	17,795	16,440	1,305	49.4	88.3	15,598	12,756	1,049

Table 6: Tracking performance of each environment on validation split. Detection model trained with and without domain-specific data are compared. Without any environment-specific data, trackers' performance drops significantly.

Method	w/Env Data	Env	Without external data					With external data				
			IDF1↑	MOTA↑	FP↓	FN↓	IDs↓	IDF1↑	MOTA↑	FP↓	FN↓	IDs↓
DMCT-TD	✗	Cafe	77.4	95.7	503	6,635	39	76.0	96.8	488	4,885	36
DMCT-TD	✓	Cafe	76.4	96.9	740	4,385	62	74.8	97.1	742	4,119	53
DMCT-TD	✗	Industry	73.8	87.7	7,400	15,661	62	74.2	90.0	7,714	11,039	67
DMCT-TD	✓	Industry	79.0	91.1	7,692	8,947	64	79.4	92.6	8,021	5,812	47
DMCT-TD	✗	Lobby	88.4	96.4	520	6,587	83	82.4	97.2	115	5,419	54
DMCT-TD	✓	Lobby	85.7	96.2	31	7,520	49	86.8	97.3	145	5,303	25
DMCT-TD	✗	Office	85.2	93.7	714	6,420	43	85.9	97.6	884	1770	38
DMCT-TD	✓	Office	81.3	97.4	787	2,182	42	89.0	98.0	994	1,237	47
DMCT-TD	✗	Retail	56.4	87.7	8,622	21,549	592	57.8	85.9	13258	21397	747
DMCT-TD	✓	Retail	58.5	89.6	5,820	19,820	403	65.3	91.3	9333	12034	395

Table 7: Tracking performance of each environment on validation split with the top-down detector. We compare the detection model trained with and without each domain-specific data, which demonstrate similar performance.

Env	Generalization		Adaptation		Supervised	
	mAP	R-1	mAP	R-1	mAP	R-1
Cafe	48.82	77.78	63.61	88.01	59.55	87.80
Industry	39.42	65.84	51.39	79.15	44.77	76.26
Lobby	46.08	72.63	60.36	87.43	51.63	82.79
Office	42.89	73.47	58.72	80.64	51.20	76.79
Retail	28.46	49.33	33.25	58.29	31.64	57.43

Table 8: Person re-identification (ReID) performance of each environment on the testing split. We report the performance of three different training settings.

we build the largest multi-camera multiple people tracking dataset with the help of an auto-annotation system, which employs various calibrated depth sensors and RGB sensors to construct a robust 3D tracker and generates reliable multi-camera tracking ground truth. Our dataset offers high-quality, dense annotations for every frame. We study the performance of two real-time trackers and one robust ReID model on our dataset. The results suggest that a large-scale dataset allows tracking systems and the ReID model to perform better. We believe these findings will benefit real-world tracking systems. For example, we can deploy one auto-annotation system, collect data and train adapted models, which will be useful for large chain retailers whose interior design are similar across stores. On the other hand, the experiments also show the challenges of designing a highly accurate multi-camera tracking system in a cluttered and crowded environment, and the baseline methods are far from meeting the accuracy requirements of the demanding applications. We hope our dataset can encourage more research efforts to be invested in this topic.

Acknowledgement We would like to thank Joe Filcik, Thomas Soemo, Yumao Lu and others in Microsoft Azure Cognitive Service team for their support.

References

[1] Amazon go. http://amazongo.com, 2017.

[2] Xavier Alameda-Pineda, Jacopo Staiano, Ramanathan Subramanian, Ligia Maria Batrinca, Elisa Ricci, Bruno Lepri, Oswald Lanz, and Nicu Sebe. SALSA: A novel dataset for multimodal group behavior analysis. *CoRR*, abs/1506.06882, 2015.

[3] Mykhaylo Andriluka, Stefan Roth, and Bernt Schiele. People-tracking-by-detection and people-detection-by-tracking. In *2008 IEEE Conference on computer vision and pattern recognition*, pages 1–8. IEEE, 2008.

[4] Pierre Baqué, François Fleuret, and Pascal Fua. Deep occlusion reasoning for multi-camera multi-target detection. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 271–279, 2017.

[5] Jerome Berclaz, Francois Fleuret, Engin Turetken, and Pascal Fua. Multiple object tracking using k-shortest paths optimization. *IEEE transactions on pattern analysis and machine intelligence*, 33(9):1806–1819, 2011.

[6] Maxim Berman, Hervé Jégou, Andrea Vedaldi, Iasonas Kokkinos, and Matthijs Douze. Multigrain: a unified image embedding for classes and instances. *arXiv preprint arXiv:1902.05509*, 2019.

[7] 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.

[8] Tatjana Chavdarova, Pierre Baqué, Stéphane Bouquet, Andrii Maksai, Cijo Jose, Timur Bagautdinov, Louis Lettry, Pascal Fua, Luc Van Gool, and François Fleuret. Wildtrack: A multi-camera hd dataset for dense unscripted pedestrian detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5030–5039, 2018.

[9] Tatjana Chavdarova and François Fleuret. Deep multi-camera people detection. In *2017 16th IEEE International Conference on Machine Learning and Applications (ICMLA)*, pages 848–853. IEEE, 2017.

[10] He Chen, Pengfei Guo, Pengfei Li, Gim Hee Lee, and Gregory Chirikjian. Multi-person 3d pose estimation in crowded scenes based on multi-view geometry. In *European Conference on Computer Vision*, pages 541–557. Springer, 2020.

[11] Weihua Chen, Lijun Cao, Xiaotang Chen, and Kaiqi Huang. An equalized global graph model-based approach for multi-camera object tracking. *IEEE Transactions on Circuits and Systems for Video Technology*, 27(11):2367–2381, 2016.

[12] Afshin Dehghan, Shayan Modiri Assari, and Mubarak Shah. Gmmcp tracker: Globally optimal generalized maximum multi clique problem for multiple object tracking. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4091–4099, 2015.

[13] Ran Eshel and Yael Moses. Homography based multiple camera detection and tracking of people in a dense crowd. In *2008 IEEE Conference on Computer Vision and Pattern Recognition*, pages 1–8. IEEE, 2008.

[14] James Ferryman and Ali Shahrokni. Pets2009: Dataset and challenge. In *2009 Twelfth IEEE international workshop on performance evaluation of tracking and surveillance*, pages 1–6. IEEE, 2009.

[15] Francois Fleuret, Jerome Berclaz, Richard Lengagne, and Pascal Fua. Multicamera people tracking with a probabilistic occupancy map. *IEEE transactions on pattern analysis and machine intelligence*, 30(2):267–282, 2007.

[16] Lingxiao He, Xingyu Liao, Wu Liu, Xinchen Liu, Peng Cheng, and Tao Mei. Fastreid: A pytorch toolbox for general instance re-identification. *arXiv preprint arXiv:2006.02631*, 2020.

[17] Yuhang He, Xing Wei, Xiaopeng Hong, Weiwei Shi, and Yihong Gong. Multi-target multi-camera tracking by tracklet-to-target assignment. *IEEE Transactions on Image Processing*, 29:5191–5205, 2020.

[18] Martin Hofmann, Daniel Wolf, and Gerhard Rigoll. Hypergraphs for joint multi-view reconstruction and multi-object tracking. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3650–3657, 2013.

[19] Yunzhong Hou and Liang Zheng. Multiview detection with shadow transformer (and view-coherent data augmentation). In *Proceedings of the 29th ACM International Conference on Multimedia*, pages 1673–1682, 2021.

[20] Yunzhong Hou, Liang Zheng, and Stephen Gould. Multiview detection with feature perspective transformation. In *European Conference on Computer Vision*, pages 1–18. Springer, 2020.

[21] Yunzhong Hou, Liang Zheng, Zhongdao Wang, and Shengjin Wang. Locality aware appearance metric for multi-target multi-camera tracking. *arXiv preprint arXiv:1911.12037*, 2019.

[22] Hung-Min Hsu, Yizhou Wang, and Jenq-Neng Hwang. Traffic-aware multi-camera tracking of vehicles based on reid and camera link model. In *Proceedings of the 28th ACM International Conference on Multimedia*, pages 964–972, 2020.

[23] Na Jiang, SiChen Bai, Yue Xu, Chang Xing, Zhong Zhou, and Wei Wu. Online inter-camera trajectory association exploiting person re-identification and camera topology. In *Proceedings of the 26th ACM international conference on Multimedia*, pages 1457–1465, 2018.

[24] Glenn Jocher, Alex Stoken, Ayush Chaurasia, Jirka Borovec, NanoCode012, TaoXie, Yonghye Kwon, Kalen Michael, Liu Changyu, Jiacong Fang, Abhiram V, Laughing, tkianai, yxNONG, Piotr Skalski, Adam Hogan, Jebastin Nadar, imyhxy, Lorenzo Mammana, AlexWang1900, Cristi Fati, Diego Montes, Jan Hajek, Laurentiu Diaconu, Mai Thanh Minh, Marc, albinxavi, fatih, oleg, and wanghaoyang0106. ultralytics/yolov5: v6.0 - YOLOv5n 'Nano' models, Roboflow integration, TensorFlow export, OpenCV DNN support, Oct. 2021.

[25] Cheng-Hao Kuo, Chang Huang, and Ram Nevatia. Inter-camera association of multi-target tracks by on-line learned appearance affinity models. In *European Conference on Computer Vision*, pages 383–396. Springer, 2010.

[26] Rainer Kümmerle, Giorgio Grisetti, Hauke Strasdat, Kurt Konolige, and Wolfram Burgard. G2o: A general framework for graph optimization. In *2011 IEEE International*

Conference on Robotics and Automation, pages 3607–3613, 2011.

[27] Hei Law and Jia Deng. Cornernet: Detecting objects as paired keypoints. In *Proceedings of the European conference on computer vision (ECCV)*, pages 734–750, 2018.

[28] Joao Paulo Lima, Rafael Roberto, Lucas Figueiredo, Francisco Simoes, and Veronica Teichrieb. Generalizable multi-camera 3d pedestrian detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1232–1240, 2021.

[29] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco: Common objects in context. In *European conference on computer vision*, pages 740–755. Springer, 2014.

[30] Hao Luo, Youzhi Gu, Xingyu Liao, Shenqi Lai, and Wei Jiang. Bag of tricks and a strong baseline for deep person re-identification. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*, pages 0–0, 2019.

[31] Wenhan Luo, Junliang Xing, Anton Milan, Xiaoqin Zhang, Wei Liu, and Tae-Kyun Kim. Multiple object tracking: A literature review. *Artificial Intelligence*, page 103448, 2020.

[32] 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.

[33] 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*, pages 6036–6046, 2018.

[34] Liangchen Song, Jialian Wu, Ming Yang, Qian Zhang, Yuan Li, and Junsong Yuan. Stacked homography transformations for multi-view pedestrian detection. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 6049–6057, 2021.

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

[36] Hanyue Tu, Chunyu Wang, and Wenjun Zeng. Voxelpose: Towards multi-camera 3d human pose estimation in wild environment. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 197–212. Springer, 2020.

[37] Minh Phuoc Vo, Ersin Yumer, Kalyan Sunkavalli, Sunil Hadap, Yaser A Sheikh, and Srinivasa G Narasimhan. Self-supervised multi-view person association and its applications. *IEEE transactions on pattern analysis and machine intelligence*, 2020.

[38] Jiuqing Wan and Liu Li. Distributed optimization for global data association in non-overlapping camera networks. In *2013 Seventh International Conference on Distributed Smart Cameras (ICDSC)*, pages 1–7. IEEE, 2013.

[39] Xiaolong Wang, Ross Girshick, Abhinav Gupta, and Kaiming He. Non-local neural networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 7794–7803, 2018.

[40] Longhui Wei, Shiliang Zhang, Wen Gao, and Qi Tian. Person transfer gan to bridge domain gap for person re-identification. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 79–88, 2018.

[41] Longyin Wen, Zhen Lei, Ming-Ching Chang, Honggang Qi, and Siwei Lyu. Multi-camera multi-target tracking with space-time-view hyper-graph. *International Journal of Computer Vision*, 122(2):313–333, 2017.

[42] Yuanlu Xu, Xiaobai Liu, Yang Liu, and Song-Chun Zhu. Multi-view people tracking via hierarchical trajectory composition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4256–4265, 2016.

[43] Yuhao Xu and Jiakui Wang. A unified neural network for object detection, multiple object tracking and vehicle re-identification. *arXiv preprint arXiv:1907.03465*, 2019.

[44] Quanzeng You and Hao Jiang. Action4d: Online action recognition in the crowd and clutter. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11857–11866, 2019.

[45] Quanzeng You and Hao Jiang. Real-time 3d deep multi-camera tracking. *arXiv preprint arXiv:2003.11753*, 2020.

[46] Amir Roshan Zamir, Afshin Dehghan, and Mubarak Shah. Gmcp-tracker: Global multi-object tracking using generalized minimum clique graphs. In *European conference on computer vision*, pages 343–356. Springer, 2012.

[47] Shu Zhang, Elliot Staudt, Tim Faltemier, and Amit K Roy-Chowdhury. A camera network tracking (camnet) dataset and performance baseline. In *2015 IEEE Winter Conference on Applications of Computer Vision*, pages 365–372. IEEE, 2015.

[48] Yifu Zhang, Chunyu Wang, Xinggang Wang, Wenyu Liu, and Wenjun Zeng. Voxeltrack: Multi-person 3d human pose estimation and tracking in the wild. *arXiv preprint arXiv:2108.02452*, 2021.

[49] Yifu Zhang, Chunyu Wang, Xinggang Wang, Wenjun Zeng, and Wenyu Liu. Fairmot: On the fairness of detection and re-identification in multiple object tracking. *International Journal of Computer Vision*, pages 1–19, 2021.

[50] Liang Zheng, Yi Yang, and Alexander G Hauptmann. Person re-identification: Past, present and future. *arXiv preprint arXiv:1610.02984*, 2016.