DESIGN AND IMPLEMENTATION OF A MULTI-TARGET MULTI-CAMERA TRACKING SOLUTION

DESIGN AND IMPLEMENTATION OF A MULTI-TARGET MULTI-CAMERA TRACKING SOLUTION

BY

RUIZHE ZHANG, B.CS.

A REPORT

SUBMITTED TO THE DEPARTMENT OF COMPUTING & SOFTWARE

AND THE SCHOOL OF GRADUATE STUDIES

OF MCMASTER UNIVERSITY

IN PARTIAL FULFILMENT OF THE REQUIREMENTS

FOR THE DEGREE OF

MASTER OF ENGINEERING

Master of Engineering (2022) McMaster University
(Computing & Software) Hamilton, Ontario, Canada

TITLE:	DESIGN AND IMPLEMENTATION OF A MULTI-TARGET MULTI-CAMERA TRACKING SOLUTION
AUTHOR:	Ruizhe Zhang
SUPERVISOR:	Dr. Rong Zheng
NUMBER OF PAGES:	vi, 30

Acknowledgements

I would like to thank my supervisor, Professor Rong Zheng, for providing me with the opportunity to conduct research under her supervision. Her guidance and advice are always valuable and helpful for my research and study. I also wish to extend my thanks to Keivan Nalaie for his patient instructions and knowledge sharing during this project. I would also like to thank Philipp Köhl, one of the original authors of WDA, for providing the MTA dataset and supporting information about the WDA paper. Many thanks to my girlfriend, Xingyu Chen, for her encouragement and care for my study and life. I am deeply grateful to my parents for their years of unconditional support.

Contents

1 Introduction **2**

1.1 Motivation and Contributions 3

1.2 Related Work 4

1.2.1 Single-Camera Tracking 4

1.2.2 Multi-Camera Tracking 5

1.2.3 Dataset for MTMCT 6

1.3 Organization of the Report 7

2 Multi-Target Multi-Camera Tracking **8**

2.1 Overview 8

2.2 Person Detection and Re-ID Features 9

2.3 Single-Camera Tracking 10

2.4 Multi-Camera Tracking 13

3 Performance Evaluation **14**

3.1 Datasets and Metrics 14

3.2 Backbone Models and Training 15

3.3 Study of Component Models and Parameter Settings 16

3.3.1 Data Association Methods 16
3.3.2 Effects of Training Data Volume 17
3.3.3 Impact of Training Data Format 18
3.3.4 Training Epochs 19
3.3.5 Other Parameter Settings 19
3.4 Overall Performance 20
3.5 Qualitative Evaluation 21

4 Conclusion and Future Works **25**

List of Figures

2.1 An overview of the proposed MTMCT pipeline [1, 2]. 9
2.2 An illustration of the tracking-by-detection paradigm [3]. 11
3.1 Sample scenes of the MTA dataset [1]. 15
3.2 Sample tracking results from an overlapping scenario. The top figure is taken from cam 2, the middle and bottom ones are taken from cam 3. Bounding boxes and identities are only marked for pedestrian no. 11 for demonstration. 23
3.3 Sample tracking results from a non-overlapping scenario. The top and middle figures are taken from cam 0 (underground), and the bottom is taken from cam 1. Bounding boxes and identities are only marked for pedestrian no. 97 for demonstration. 24

Abstract

Multi-Target Multi-Camera Tracking (MTMCT) aims to continuously track multiple targets across video streams from multiple cameras of interest. It is playing an increasingly important role in computer vision applications such as visual surveillance, crowd behaviour prediction, and sports analysis. To date, most MTMCT methods accomplish their tasks by first detecting targets and forming tracks within each single camera assisted by the detections and extracted re-identification (re-ID) features which then match single-camera tracks across all cameras in order to generate complete trajectories. Many existing MTMCT methods mainly focus on the multi-camera track-matching part and rely on two-stage single-camera trackers, which consist of two neural network models for object detection and feature extraction, to form single-camera tracks. However, the inference time of those solutions is usually slow, as the two models do not share features, and the re-ID model needs to be applied on each bounding box detected by the detector model. The training of two separate models also requires both extra data and time. We propose the design and implementation of an effective and efficient MTMCT solution by adopting a state-of-the-art single-shot tracker, an effective data association method, and a weighted aggregation hierarchical clustering approach. The resulting solution outperforms the original methods on a high-resolution simulated dataset and achieves a balance between performance and inference speed.

Chapter 1

Introduction

This report presents our design and implementation of a Multi-Target Multi-Camera Tracking (MTMCT) solution. MTMCT aims to determine the trajectories of multiple targets of interest across multiple camera angles [4]. MTMCT is an important task in computer vision due to an increasing need in applications such as visual surveillance, crowd behaviour prediction, and sports analysis [5].

Valued as it is, MTMCT is also a challenging task. It not only involves numerous sub-tasks such as single-camera Multi-Object Tracking (MOT) and Re-Identification (re-ID), but also needs to handle specific cross-camera problems. For instance, targets may look significantly different under various lighting conditions and pose changes across multi-cameras [6]. Choosing the suitable datasets that cover the above-mentioned challenging characteristics, while at the same time not violating privacy protocols for the study is also essential. In this report, we present an MTMCT solution based on the tracking-by-detection paradigm and cross-camera data association. We use a high-resolution, simulated, and synchronized video dataset [7] to conduct experiments.

Section 1.1 discusses our motivation to conduct research on MTMCT as well as the contributions we make in this work. Section 1.2 reviews some related works on MTMCT. And lastly, Section 1.3 summarizes the organization of the rest of this report.

1.1 Motivation and Contributions

Traditionally, multi-camera tracking is done in two phases: generating single-camera tracks for detected targets, and matching tracks across different camera views to form complete multi-camera tracks [4, 7]. Many MTMCT works use two separate deep learning models for the first phase, and mainly focus on the track matching part [1, 7, 8, 9, 10]. However, inference time is slow as the two models do not share features, and instead of re-using the backbone features, the re-ID model needs to be applied on each bounding box located by the detector model [11]. Additionally, the training of two separate network models also requires extra data and time.

Even though there are faster single-shot tracking methods [11, 12] that estimate objects and extract re-ID features using a single model, they are rarely being applied in MTMCT solutions. In order to achieve both effectiveness and efficiency on MTMCT, we modify and integrate a fast single-shot MOT tracker [11] into an MTMCT framework [1] to form a new solution. Our contributions include:

- The proposed solution outperforms the original single-shot tracker on single-camera tracking by utilizing a more effective data association method.
- The proposed solution also outperforms the original MTMCT framework on both single and multi-camera tracking by replacing both the detection and the

re-ID feature extraction models with a single-shot model.

- The proposed solution reduces the inference time by around 50% when compared to the original MTMCT framework.

1.2 Related Work

We first review some related works, including both single-camera and multi-camera tracking works, as multi-camera tracking solutions build upon single-camera ones. Then, we review some datasets for MTMCT.

1.2.1 Single-Camera Tracking

Single-camera trackers can be classified as either two-stage trackers or single-shot trackers. Traditionally, person detection and re-ID feature extraction are done by two separate models [7, 13, 14]. In addition to those two models, TPAGT [15] also adopts an adaptive graph neural network (AGNN) to fuse locations, appearances and historical information in order to help distinguish different detected objects. Although TPAGT provides state-of-the-art performance, the architecture is too complex, and using three models would increase inference time. Tracktor [16], on the other hand, pushes the tracking-by-detection paradigm to its limit, using only an object detector to perform the task of tracking via bounding-box regression. However, in order to achieve state-of-the-art tracking performance, Tracktor still needs to add a re-ID extension consisting of a separate neural network model.

The complex designs of the above methods provide good performance at the cost of longer inference time. Single-shot trackers, on the other hand, detect objects

and extract re-ID features using a single network, and offer the benefit of reduced computation complexity [11, 12]. In order to reduce inference time, existing single-shot trackers including Track R-CNN [12] and JDE [17] are modified from anchor-based object detectors such as YOLO [18] and reuse existing backbone features for re-ID. However, the tracking accuracy usually drops when compared to the two-stage methods, as the task of re-ID is not fairly treated, resulting in good detection results but high ID switches [11].

In particular, the FairMOT designed by Zhang *et al.* [11] balances the detection and re-ID tasks by using a single network consisting of two branches to predict object locations and extract re-ID features jointly. More specifically, the two branches estimate object centers with an anchor-free method, and homogeneously estimate re-ID features for each pixel. It therefore learns high-quality re-ID features, and not only reduces inference time, but also achieves state-of-the-art performance.

We adopt FairMOT [11] into the proposed solution as a replacement to the two-stage modules from a MTMCT framework to generate single-camera detection bounding boxes and appearance features simultaneously.

1.2.2 Multi-Camera Tracking

Most MTMCT methods utilize object detection, re-ID feature extraction, and track association as the key building blocks of their solutions [1, 3, 4, 8, 9]. Ristani *et al.* [4] emphasize the correlation between tracking and re-ID features, therefore using an adaptive weighted triplet loss during the re-ID network model training to enhance the training effectiveness on hard feature training samples. They also use hierarchical reasoning to first form short tracklets, and then group them into longer

single-camera and, eventually, multi-camera trajectories. Tesfaye *et al.* [6] use similar strategies and, more specifically, apply clustering techniques that not only associate tracks from different cameras, but also link multiple wrongly separated single-camera tracks belonging to the same identity.

Among the MTMCT solutions with similar strategies mentioned above, ones with a modular design such as the Weighted Distance Aggregation (WDA) tracker published by Köhl *et al.* [1] provide the convenience of exchanging components, including the person detection module, the feature extraction module, and the single-camera tracker. Aside from the modular design, WDA also integrates a core track comparison component that computes five different feature distances between tracks. Then, a hierarchical clustering approach merges tracks with a weighted aggregation of all the distances.

We use WDA as the basis of our solution as it gives us a complete MTMCT pipeline and provides space for improvements and customization. Different from the original WDA, we exchange the individual detection and feature extraction modules with a single-shot network model from FairMOT [11] to increase inference time and improve performance.

1.2.3 Dataset for MTMCT

Dataset is another critical component in MTMCT research, as the performance of both detection and feature extraction network models highly rely on training data. The DukeMTMC dataset [19] is widely used in tracking research, as it provides high-definition, long-lasting videos that are recorded by eight cameras and contain 2000+ identities. However, it only covers outdoor scenarios, wherein light and weather

conditions are relatively constant. Most importantly, the dataset is then discontinued following a privacy investigation. The Multi-Camera Track Auto (MTA) dataset created by Köhl *et al.* [1] makes for a suitable alternative, even though it is comprised of simulated videos. MTA provides high-quality videos for MTMCT and contains various illumination and weather changes in both indoor and outdoor regions. We use MTA as the primary training and validation dataset for our research.

1.3 Organization of the Report

The rest of the report is organized as follows. Chapter 2 introduces the methodologies and technical details of our MTMCT solution. Chapter 3 presents study of component models and parameter settings, evaluations, and comparisons with other baseline methods. Chapter 4 concludes our works and lists potential future works.

Chapter 2

Multi-Target Multi-Camera Tracking

In this chapter, we present an overview of the proposed pipeline and the technical details and methodologies behind the individual components, including person detection, re-ID feature extraction, single-camera tracking, multi-camera track distance calculation, and clustering.

2.1 Overview

Most recent MTMCT solutions consist of two phases; a single-camera tracking phase and a cross-camera track aggregation phase which forms complete trajectories from multi-camera views [7]. Our solution, as shown below in Fig.2.1, follows this pattern and focuses primarily on optimizing the performance of the single-camera tracking phase, as the overall multi-camera tracking performance greatly depends on the formed single-camera tracks. Our method takes the input videos and processes

them frame by frame. Frames from each camera are first passed to a single-shot MOT network for object detection and feature extraction. Then, after a data association step, pairwise distances between the formed single-camera tracks will be calculated with weighted distance aggregation of multiple distances. Finally, multi-camera tracks are generated by hierarchical clustering.

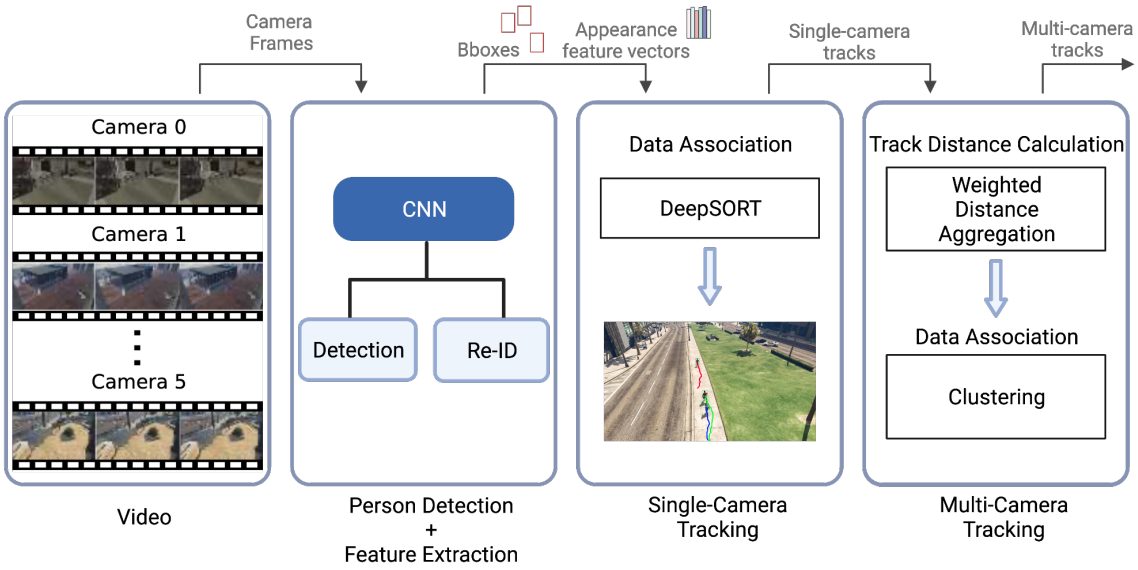

Figure 2.1: An overview of the proposed MTMCT pipeline [1, 2].

2.2 Person Detection and Re-ID Features

Object detection is essential in computer vision, as it is the foundation of other advanced tasks including face detection, autonomous driving, and tracking. Depending on the specific application scenario, the class of objects to be detected can belong to different categories such as pedestrians, vehicles, and animals. In this report and the corresponding works, we consider only pedestrian tracking. The outputs take the form of rectangular bounding boxes which specify the locations of the detected

targets. Instead of a separate detector model, we utilize a single-shot network, also used for extracting re-ID features, for estimating positions of the targets.

On top of the detected bounding boxes, re-ID features can enhance association and the resulting tracking performance. Traditionally, images are cropped based on the detected bounding boxes and fed to a re-ID embedding network to extract appearance features [11]. Those features are then used to link individual bounding boxes in order to form tracks.

Many existing works use two separate models for object detection and feature extraction [7, 13, 14, 15]. In order to enhance the efficiency of our solution, we instead use a single-shot tracker, FairMOT [11], to localize objects and extract their re-ID features simultaneously with a single two-branch neural network model. The smaller re-ID feature dimension (128) of FairMOT also helps to reduce computation time during inference when compared to other traditional two-stage trackers.

2.3 Single-Camera Tracking

Like many state-of-the-art works, we follow the commonly used tracking-by-detection paradigm as illustrated in Fig.2.2, i.e., to solve the issue of tracking as a data association problem. More specifically, it associates detected bounding boxes across frames in a video sequence to form single-camera tracks [20].

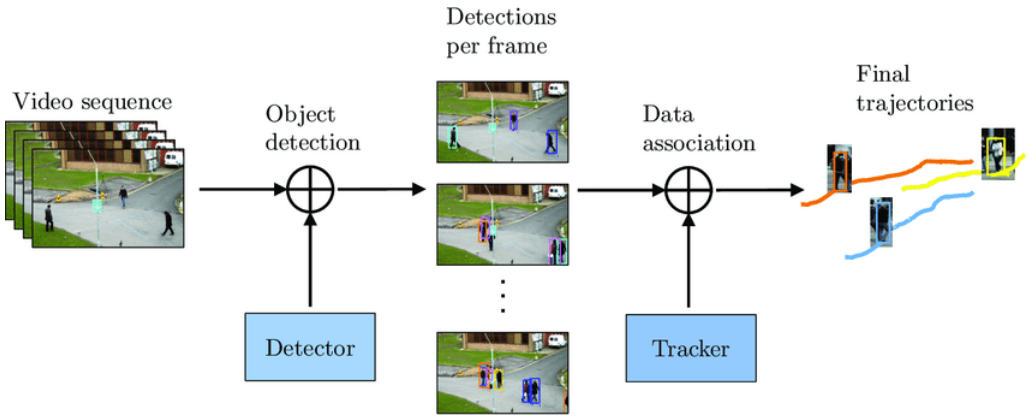

Figure 2.2: An illustration of the tracking-by-detection paradigm [3].

In order to link detected bounding boxes in each frame to existing tracks, two association steps are done using different distance metrics. Firstly, the cosine distances computed on re-ID features are used to form a distance matrix between existing tracks and new detections [11]. Note that the Kalman Filter is also used here to update the cost matrix:

- **Kalman Filter:** Kalman Filter is an algorithm that can predict future positions of objects based on current positions [21]. We use it to predict the locations of existing tracks in the current frame and then compare them to the detected bounding boxes. If the distances between the detected bounding boxes and predicted positions exceed the specified threshold, then the detections will not be linked to the existing tracks [11]. The corresponding cost will be set to infinity, which prevents the detections with large motions from being linked in the Hungarian Algorithm.

Other than re-ID features, Intersection over Unions (IoU) of the bounding boxes is also used for computing a cost matrix:

- **Intersection over Unions (IoU):** IoU is an evaluation metric that measures the intersection ratio between the area of the bounding box from the last frame in an existing track and the area of a newly-detected bounding box in the current frame [22]. The formula is as follows:

$$IoU = \frac{AreaOfOverlap}{AreaOfUnion}. \tag{2.1}$$

Both Kalman Filter and IoU calculations have a threshold wherein associations with costs higher than this value are disregarded. After computing each cost matrix, the classical yet efficient Hungarian algorithm is then used for linking detected boxes to tracks:

- **Hungarian Algorithm:** The Hungarian Algorithm links new detection by treating it as a bipartite graph optimal matching problem [20] between the detected bounding boxes in the current frame and existing tracks. Each edge represents the association between a detected bounding box and an existing track, and the weight is the corresponding cost in the computed cost matrix. The detected boxes will be linked to existing tracks with the minimum distance. Unmatched tracks are marked as missed, while the unmatched detected boxes initiate new tracks.

As shown in Fig.2.1, we utilize the detected bounding boxes and appearance features from the two-branch network model of FairMOT, and feed them to the data association part of DeepSORT, the single-camera tracker of WDA [1]. Although both FairMOT and DeepSORT essentially apply the above tracking techniques, DeepSORT does apply more optimized thresholds for IoU and Kalman Filter, as well as

more matching factors such as track age [1, 14].

By applying a more efficient, single-shot network from FairMOT for person detection and feature extraction, and a better track association mechanism from DeepSORT, we are able to form more accurate single-camera tracks than both original trackers, which are then used in the subsequent multi-camera phase.

2.4 Multi-Camera Tracking

In order to link single-camera tracks to complete trajectories over multi-camera, we first need to group tracks that belonging to the same identity together. By using the WDA [1] method, we compute five different feature distances between tracks, and then aggregate the distances by linearly combining them with respective weights. Note that said aggregation provides a distance metric for track comparison while helping correct fragmented, single-camera tracks caused by occlusions or temporary absence from the camera view. Furthermore, two tracks that violate certain constraints, such as one pedestrian cannot appear in multiple tracks in one camera view at the exact moment, are assigned infinity distance and therefore not linked in the subsequent clustering step.

Subsequently, tracks are merged based on the aggregated distance using agglomerative hierarchical clustering. More specifically, every single track is initially a separate cluster. The two closest clusters are merged every step until a distance threshold, or a desired number of clusters has been reached. The final resulting clusters are complete, multi-camera trajectories.

Chapter 3

Performance Evaluation

This chapter introduces the datasets and metrics used in the evaluation and the implementation details for the experiments. Also presented are studies of the impact of individual design decisions on the performance of our method. An overall evaluation and comparison with other baselines, followed by a qualitative evaluation, are presented last.

3.1 Datasets and Metrics

The MTA dataset [1] is used as the major training and testing dataset. MTA is a large-scale simulated dataset including 1920x1080 video scenes recorded at 41 FPS by six cameras, and includes overlapping and non-overlapping camera views within indoor and outdoor scenes. Fig.3.1 shows sample images from all six camera views. The total video length of the MTA dataset is more than 10 hours, and covers various crowd density and light conditions, making it a suitable evaluation dataset for MTMCT. By transforming the MTA dataset to match MOT17 annotations, it is

now more accessible for MOT trackers. MOT17 [23] and COCO [24] are also used for pre-training.

All results are based on average evaluations of tracking results from the six cameras. We use selected MTA's built-in metrics as evaluation metrics [1]. More specifically, the MOTA score is selected as the target metric, which measures the overall tracking accuracy of multiple objects. IDF1 is also used in order to evaluate the effectiveness of ID matching, which is the ratio of correctly-identified detections over the average number of ground-truth and computed detections [19]. Similarly, IDs that stand for the number of identity switch errors are also compared. Mostly-Tracked (MT) and Mostly-Lost (ML) are also displayed in the final evaluation section to give a more comprehensive assessment.

Figure 3.1: Sample scenes of the MTA dataset [1].

3.2 Backbone Models and Training

We use [11] for both person detection and re-ID feature extraction, therefore their baseline, a DLA-34 [25] network pre-trained on the COCO dataset [24], is used as

the backbone model. We also pre-train the models with MOT17 in the following experiments, but not for the final comparison. We use default training configurations of FairMOT [11] and train all models for 30 epochs on two Tesla P100 GPUs.

3.3 Study of Component Models and Parameter Settings

In this section, we present experiments and evaluations of the impact of different design decisions and training strategies of our approach.

3.3.1 Data Association Methods

We study the impact of various combinations of detector/feature extractor and data association approaches on single-camera tracking performance. More specifically, the performances of three combinations are compared, including feeding detected boxes and features from WDA [1] to FairMOT [11], and vice versa, as well as the original FairMOT construction.

The comparisons are shown in Table 3.1. The second row shows that by feeding WDA's detection and features to FairMOT [11], both MOTA and IDF1 increase slightly. However, the IDs are much higher, showing that the WDA's [1] detection is effective, but the re-ID features cause increased mismatch errors. The best performances with 59.0% MOTA, 42.2% and 3224 IDs are achieved when feeding FairMOT's detection and features to WDA. The results demonstrate that WDA's data association method, i.e., DeepSORT [14], is more effective than the original method of FairMOT.

Table 3.1: Comparison of different combinations of detector & feature extractor and data association component.

Detector & Feature Extractor	Data Association	MOTA↑	IDF1↑	IDs↓
FairMOT	FairMOT	53.1	38.4	803.3
WDA	FairMOT	56.9	41.0	995.7
FairMOT	WDA	**59.0**	**42.2**	**537.3**

3.3.2 Effects of Training Data Volume

In order to verify whether more training data affects the performance of our method, models trained on two datasets with different overall lengths were evaluated.

Our preliminary result was not ideal when compared to WDA's baseline performance, as we only trained our model with a tiny portion, i.e., two-minute-long videos from each of the six camera views out of the entire dataset, while WDA used all available training data.

We compare the original two-minute video sets with a newly made training dataset consisting of a 4-minute-long video for each camera view. Because of the simulation nature and faster time lapses, longer MTA videos cover more significant light conditions and crowd density changes. Therefore, for each camera view, we take four 1-minute-long portions from sections of the full MTA video covering different illumination conditions and combine them into one 4-minute-long video, instead of simply taking a consecutive 4-minute-long portion. As shown in Table 3.2, with more training data, the performances on both single and multi-camera tracking tasks increase. Due to computational limitations, we can only utilize such an amount of data. However, it shows that there is room for improvements in our model's performance with the addition of more training data.

Table 3.2: Impact of increasing training data size on both single and multi-camera performance.

Trained on	Single-Camera			Multi-Camera		
	MOTA↑	IDF1↑	IDs↓	MOTA↑	IDF1↑	IDs↓
MTA 2 min×6	53.1	38.4	803.3	50.1	23.6	851.5
MTA 4 min×6	**58.7**	**40.4**	**796.0**	**53.0**	**26.6**	**835.2**

3.3.3 Impact of Training Data Format

We aim to study the impact of different formats of images frames on the performance of our model. Given the same video sets, the divided image frames used for training can be in different image formats. The method is evaluated trained on images frames divided from the same set of videos, but in two different formats, a lossy compressed format JPEG, and the PNG format, which supports lossless data compression.

As shown in Table 3.3, the model trained on PNG format images outperforms the one trained by JPEG images, demonstrating that the compression loss affects the training effectiveness given the identical resolutions. PNG images do require more storage spaces as they are lossless compressed. However, the training speed does not increase based on that factor, as the resolution and the processed pixel vectors remain the same dimensions.

Table 3.3: Impact of using different training image formats on single-camera tracking performance.

Trained on	MOTA↑	IDF1↑	IDs↓
MTA image frames in JPEG	58.6	46.0	683.8
MTA image frames in PNG	**64.1**	**48.0**	**588.2**

3.3.4 Training Epochs

We also study the impact of different training epochs on both our single and multi-camera tracking performances. The models are compared trained on the same data with 30 epochs, the default setting of FairMOT [11], and a lower number of 20 epochs, respectively.

As displayed in Table 3.4, the model trained with 20 epochs performs slightly better than that trained with 30 epochs on both MOTA and IDF1 scores. The model trained with 30 epochs provides fewer IDs in both single and multi-camera tracking phases. The results demonstrate that on the MTA dataset [1], the model does not need extra numbers of training iterations in order to achieve optimized overall tracking performance. However, more training epochs can slightly help prevent ID switches.

Table 3.4: Impact of different training epochs on tracking performance.

No. of Training Epochs	Single-Camera			Multi-Camera		
	MOTA↑	IDF1↑	IDs↓	MOTA↑	IDF1↑	IDs↓
20	**70.8**	**47.8**	470.2	**65.6**	**31.5**	494.5
30	70.1	47.3	**447.2**	65.3	26.8	**480**

3.3.5 Other Parameter Settings

We also study the impacts of thresholds on accepting detected bounding boxes, such as different minimum bounding box sizes, and whether to include vertical checks on the box shapes. Both of these changes resulted in worse performance. Additionally, we use the best-weight finding function from WDA [1] to find out the best weights for the weighted distance aggregation logic, though the resulting weights only bring minor improvements (less than 1%) to the overall performance.

3.4 Overall Performance

The following presents an overall evaluation of our approach and compares it with two baseline methods, namely the original FairMOT [11], and the WDA tracker [1]. We compare with FairMOT only on single-camera tracking and WDA on both single and multi-camera tracking. In addition to tracking metrics, inference speeds in frames per second (FPS) are also documented. All the tests were conducted on a single Tesla P100 GPU.

As shown in Table 3.5, our solution's overall single-camera tracking performance is superior to the other two methods except for the similar IDF1 score of FairMOT [11]. Speed-wise, our solution surpasses WDA [1] but is slower than FairMOT. In the current implementation, detections and re-ID features are taken from FairMOT and then fed to WDA's DeepSORT [14] tracker for data association, which takes extra time. In terms of multi-camera tracking, our solution outperforms the original WDA by a large margin. Speed-wise, we also shortened the time by 50%. Note that both FairMOT and WDA's detector models are trained on the same dataset, i.e., a set of six 2-minute-long videos selected from the full MTA video. All image frames used for training are in PNG format. For WDA's re-ID model, we directly use the best-trained baseline model shared by the authors [1], as the training data used for re-ID training is not available. Although slightly unfair to our solution, we still overcome their approach trained on a complete full-length MTA dataset which is dozens of times longer than our training data. In this sense, the proposed model outperforms their best trained model for re-ID with much less training data.

the correct tracking result after the pedestrian exits the camera field of view and re-appears in camera 1, shown in the bottom picture. It shows that the proposed solution can track pedestrians through non-overlapped camera views. However, errors such as ID switches still occur, which shows that the model still has room for improvement.

Further demonstrative videos can be found at `https://youtu.be/lS9YvbrhOdo`.

Figure 3.2: Sample tracking results from an overlapping scenario. The top figure is taken from cam 2, the middle and bottom ones are taken from cam 3. Bounding boxes and identities are only marked for pedestrian no. 11 for demonstration.

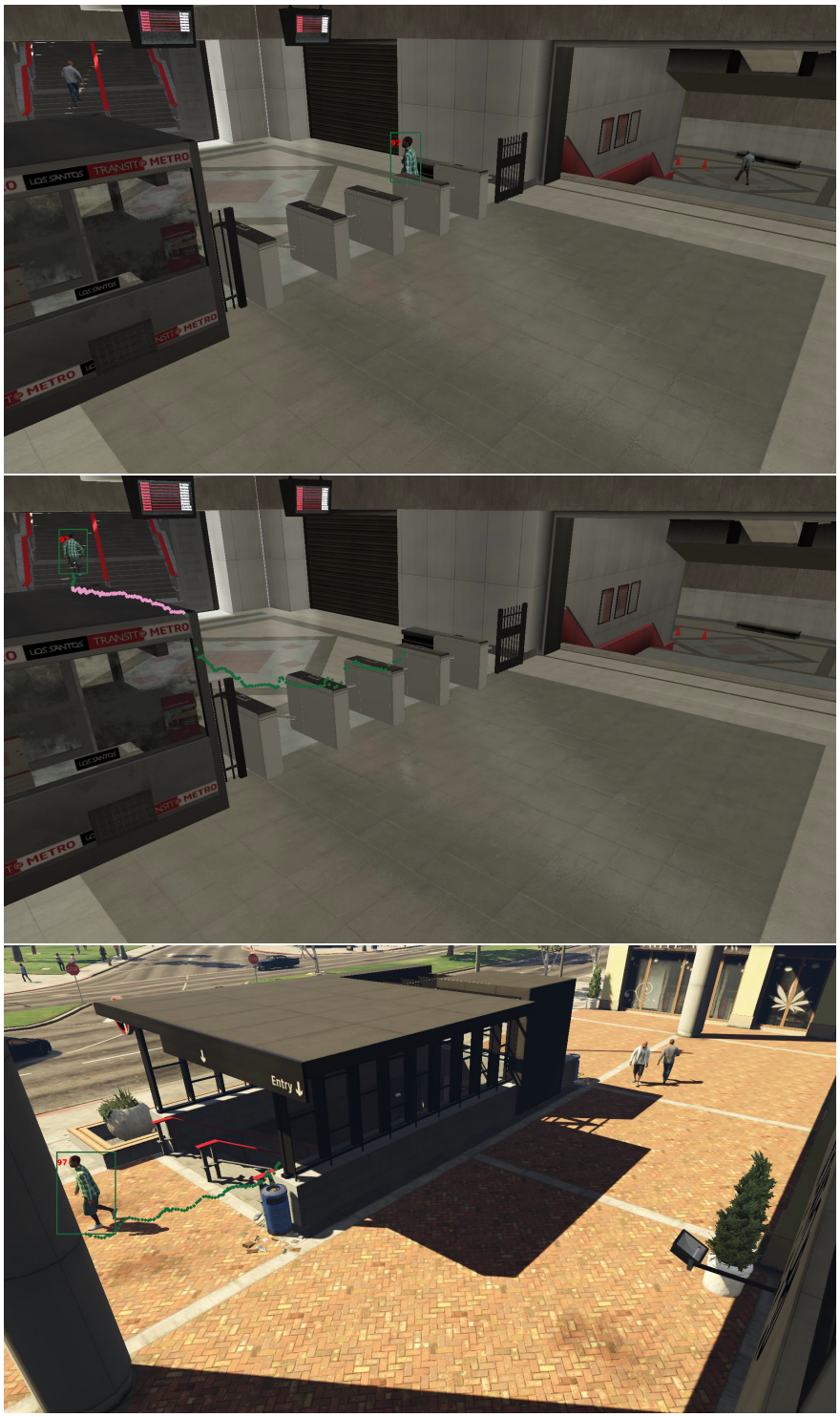

Figure 3.3: Sample tracking results from a non-overlapping scenario. The top and middle figures are taken from cam 0 (underground), and the bottom is taken from cam 1. Bounding boxes and identities are only marked for pedestrian no. 97 for demonstration.

Chapter 4

Conclusion and Future Works

In conclusion, design and implementation of this MTMCT approach was achieved by combining the single-shot detector and feature extractor network from FairMOT [11], as well as the data association component, and the weighted distance aggregation and hierarchical clustering from WDA [1]. By conducting an evaluation based on the simulated MTA dataset, we demonstrated that our solution outperformed the original methods under fair conditions and achieved a balance between performance and inference speed.

There is still room for improvement on multiple aspects, and further experiments are expected to be conducted in the future.

- On the training side, due to computational resource limitations, we only utilized a tiny portion of the full-length MTA dataset for model training, and therefore did not achieve optimal performance.

- All experiments were done based on the simulated dataset. However, tests on

real-life videos can be performed to see how effective the solution is on cross-domain person tracking.

- Implementation-wise, the combination of two MOT trackers can be improved upon. The feeding of detected boxes and extracted features can be done more coherently, which would also potentially help shorten inference time.

- Real-time track management can be added in order to enable our solution for online inference.

Bibliography

[1] Philipp Köhl, Andreas Specker, Arne Schumann, and Jürgen Beyerer. The mta dataset for multi target multi camera pedestrian tracking by weighted distance aggregation. *IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops*, 2020-June:4489–4498, 6 2020.

[2] Created with BioRender.com.

[3] Laura Leal-Taixé. Multiple object tracking with context awareness doctoral thesis.

[4] Ergys Ristani and Carlo Tomasi. Features for multi-target multi-camera tracking and re-identification. *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 6036–6046, 2018.

[5] Wenhan Luo, Junliang Xing, Anton Milan, Xiaoqin Zhang, Wei Liu, Xiaowei Zhao, and Tae-Kyun Kim. Multiple object tracking: A literature review. 9 2014.

[6] Yonatan Tariku Tesfaye, Eyasu Zemene, Andrea Prati, Marcello Pelillo, and Mubarak Shah. Multi-target tracking in multiple non-overlapping cameras using constrained dominant sets. 6 2017.

[7] 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.

[8] Michael Bredereck, Xiaoyan Jiang, Marco Körner, and Joachim Denzler. Data association for multi-object tracking-by-detection in multi-camera networks. In *2012 Sixth International Conference on Distributed Smart Cameras (ICDSC)*, pages 1–6, 2012.

[9] Kuan Wen Chen, Chih Chuan Lai, Pei Jyun Lee, Chu Song Chen, and Yi Ping Hung. Adaptive learning for target tracking and true linking discovering across multiple non-overlapping cameras. *IEEE Transactions on Multimedia*, 13:625–638, 8 2011.

[10] Kyujin Shim, Sungjoon Yoon, Kangwook Ko, and Changick Kim. Multi-target multi-camera vehicle tracking for city-scale traffic management. *IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops*, pages 4188–4195, 6 2021.

[11] Yifu Zhang, Chunyu Wang, Xinggang Wang, Wenjun Zeng, and Wenyu Liu. Fairmot: On the fairness of detection and re-identification in multiple object tracking. 4 2020.

[12] Paul Voigtlaender, Michael Krause, Aljosa Osep, Jonathon Luiten, Berin Balachandar Gnana Sekar, Andreas Geiger, and Bastian Leibe. Mots: Multi-object tracking and segmentation. *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2019-June:7934–7943, 6 2019.

[13] Yifu Zhang, Peize Sun, Yi Jiang, Dongdong Yu, Zehuan Yuan, Ping Luo, Wenyu Liu, and Xinggang Wang. Bytetrack: Multi-object tracking by associating every detection box. 10 2021.

[14] Nicolai Wojke, Alex Bewley, and Dietrich Paulus. Simple online and realtime tracking with a deep association metric. *Proceedings - International Conference on Image Processing, ICIP*, 2017-September:3645–3649, 3 2017.

[15] Chaobing Shan, Chunbo Wei, Bing Deng, Jianqiang Huang, Xian-Sheng Hua, Xiaoliang Cheng, and Kewei Liang. Tracklets predicting based adaptive graph tracking. 10 2020.

[16] Philipp Bergmann, Tim Meinhardt, and Laura Leal-Taixe. Tracking without bells and whistles. *Proceedings of the IEEE International Conference on Computer Vision*, 2019-October:941–951, 10 2019.

[17] Zhongdao Wang, Liang Zheng, Yixuan Liu, Yali Li, and Shengjin Wang. Towards real-time multi-object tracking. *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 12356 LNCS:107–122, 9 2019.

[18] Joseph Redmon, Santosh Divvala, Ross Girshick, and Ali Farhadi. You only look once: Unified, real-time object detection. *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2016-December:779–788, 12 2016.

[19] Ergys Ristani, Francesco Solera, Roger Zou, Rita Cucchiara, and Carlo Tomasi. Performance measures and a data set for multi-target, multi-camera tracking.

Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics), 9914 LNCS:17–35, 9 2016.

[20] Alex Bewley, Zongyuan Ge, Lionel Ott, Fabio Ramos, and Ben Upcroft. Simple online and realtime tracking. *Proceedings - International Conference on Image Processing, ICIP*, 2016-August:3464–3468, 2 2016.

[21] Greg Welch and Gary Bishop. An introduction to the kalman filter.

[22] Hui Xu, Chaochuan Fu, Shunyu Yao, and Xinlu Zong. An improved k-means algorithm based on intersection over union for network security. *2019 IEEE 11th International Conference on Communication Software and Networks, ICCSN 2019*, pages 514–517, 6 2019.

[23] Anton Milan, Laura Leal-Taixé, Taix´ Taixé, Ian Reid, Stefan Roth, and Konrad Schindler. Mot16: A benchmark for multi-object tracking. 3 2016.

[24] 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. *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 8693 LNCS:740–755, 5 2014.

[25] Xingyi Zhou, Dequan Wang, and Philipp Krähenbühl. Objects as points. 4 2019.