6-18-2024

Two-Step Hierarchical Multi-Camera People Tracking

Eerina Haque

Eric Huang

Sihang Li

SANTA CLARA UNIVERSITY

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

Date: June 18, 2024

I HEREBY RECOMMEND THAT THE THESIS PREPARED UNDER MY SUPERVISION BY

Eerina Haque
Eric Huang
Sihang Li

ENTITLED

Two-Step Hierarchical Multi-Camera People Tracking

BE ACCEPTED IN PARTIAL FULFILLMENT OF THE REQUIREMENTS FOR THE DEGREE OF

BACHELOR OF SCIENCE IN COMPUTER SCIENCE AND ENGINEERING

David C. Anastasiu

Digitally signed by David C. Anastasiu
Date: 2024.06.18 09:50:43 -04'00'

Thesis Advisor

Department Chair

Two-Step Hierarchical Multi-Camera People Tracking

by

Eerina Haque
Eric Huang
Sihang Li

Submitted in partial fulfillment of the requirements
for the degree of
Bachelor of Science in Computer Science and Engineering
School of Engineering
Santa Clara University

Santa Clara, California
June 18, 2024

Two-Step Hierarchical Multi-Camera People Tracking

Eerina Haque
Eric Huang
Sihang Li

Department of Computer Science and Engineering
Santa Clara University
June 18, 2024

ABSTRACT

The possibility of an efficient and accurate solution for multi-camera people tracking (MCPT) is enabled by the improvement of computing power and the advancement of machine learning technologies. The problem of multi-camera people tracking serves as a cornerstone of real-world applications such as video surveillance or warehouse automation. The current solutions for MCPT suffer from problems such as appearance inconsistency, object occlusion, etc. Our work targets tackling the challenges faced by modern MCPT algorithms to bring a more robust, efficient, and accurate solution.

ACKNOWLEDGMENTS

Table of Contents

List of Figures

Chapter 1

Introduction

With the availability of low-cost, high-frame-rate cameras, it's easier to deploy complex networks of cameras to monitor large areas in public, retail, and industrial settings to ensure safety and optimize processes. However, with all of the footage these cameras capture, it's humanly impossible to analyze every frame. Therefore, automated people-tracking solutions have been a growing area of interest in computer vision research. Multi-Camera People Tracking (MCPT) solutions monitor pedestrians using multiple cameras to uniquely identify each pedestrian and track their movements.

There are many use cases for MCPT solutions. They can ensure safety by identifying theft in retail stores. NVIDIA is helping the global retail industry tackle the $100 billion loss of goods problem due to theft, damage, and misplacement using its Multi-Camera Tracking AI workflow to anonymously track shoppers [1]. MCPT can also automate processes such as checkout in retail stores. In 2021, 7-Eleven created a patented people tracking system that tracks people and any items they leave the store holding. The tracked customer is charged for the items without the conventional checkout process [2]. Navigine, a provider of tracking solutions, offers a warehouse and logistics tracking system that uses MCPT for staff workflow optimization and safety protocol violations [3].

1.1 Problem Statement

Multi-camera people tracking (MCPT) is the computer vision (CV) task of detecting, uniquely identifying, and tracking pedestrians as they move in an area monitored by a network of multiple cameras. Given video footage from different cameras monitoring pedestrians in a scene, an MCPT solution pipeline must output trajectories, which are composed of a) a sequence of bounding boxes following the movement of each unique pedestrian across all videos, and b) an associated global ID that uniquely identifies each pedestrian across all cameras.

MCPT is an especially difficult task because it's composed of a sequence of multiple CV subtasks, namely:

1. **Detection**: Detect pedestrians from the videos in each camera and generate bounding boxes around each detected pedestrian

2. **Feature extraction**: Extract distinguishing attributes or features (e.g. appearance, location, pose) of each detected pedestrian

3. **Inter-camera association**: Associate the frames of each unique pedestrian across all cameras into *trajectories*[1] using the extracted features

4. **Re-identification**: Re-identify pedestrians when they exit/enter cameras, and when they are occluded by other pedestrians and/or objects

MCPT faces similar challenges compared to other Computer Vision (CV) tasks. Firstly, it is susceptible to ID switching. ID switching occurs when the MCPT solution struggles to distinguish similar-looking pedestrians so it incorrectly switches their global IDs. MCPT solutions also struggle with occlusion, which is when parts of the pedestrian's body cannot be seen to gather enough information to identify the pedestrian. There is also a lack of data for MCPT solutions to learn observed patterns through training so that they can track and identify pedestrians. Neural networks underlie many MCPT solutions, so a lack of data can result in insufficient information to make accurate predictions of the trajectories of pedestrians. As a result, these networks often train on synthetic data, such as simulated pedestrians in digital environments, to predict the location of objects in real camera footage. Unlike other CV tasks, variations in environmental conditions across cameras are a challenge for MCPT solutions because the appearance of each pedestrian can be drastically dissimilar in different cameras due to changes in lighting conditions and camera angles, hindering pedestrian re-identification.

In our thesis, we look specifically at solving MCPT for use cases where people are tracked indoors, which lends itself to a particular set of challenges including higher rates of object occlusion and difficult re-identification. It's more difficult to track people indoors because there tend to be more occluding objects in the monitoring area, people change their trajectories more frequently when compared to tracking people outdoors, and there are more variations in environmental conditions across cameras. As people walk out of a camera's view, computer vision models struggle to re-identify the person as they walk into the view of another camera or even the same camera.

1.2 Project Proposal

There's been plenty of research in this domain tackling these challenges but MCPT is still not a perfect technology. Our solution will be submitted to the AI City Challenge where our project will be evaluated by competing with teams from all over the world. The Higher Order Tracking Accuracy (HOTA) score will be used as the evaluation metric for the challenge, which is defined to evaluate the proportion of objects that are correctly detected and identified. Besides HOTA which is used by the AI City competition, other measures should also be taken into account such as

[1] A trajectory describes how a pedestrian moves in a monitored area across multiple cameras. It is a collection of sequential bounding boxes capturing a unique pedestrian across several frames in each camera. The sequential bounding boxes are given a consistent ID in footage across all cameras to uniquely identify each pedestrian.

speed and the number of unique IDs detected because these are factors that cannot be overlooked in reality. We aim to improve upon the anchor-guided clustering and spatio-temporal consistency solution proposed by Huang et al [4] by intelligently sampling frames to feed the MCPT solution, improving re-identification of pedestrians across frames by considering the pose similarity, and employing more sophisticated SCPT methods. With these adjustments, we increase the solution's robustness to noise and heavy occlusions, especially in real-world scenarios.

Chapter 2

Project Requirements

2.1 Functional Requirements

Because we are adopting a Two-Step Hierarchical Approach to MCPT, our solution must be compatible with the existing algorithms for object detection and single-camera tracking that we will be incorporating into our solution. As a result, our solution pipeline must fit the expected inputs and outputs of the existing algorithms we will be leveraging.

Our project also must convert our results into the expected submission format specified by AI City Challenge, so that AI City Challenge can evaluate our results. Unlike past year's challenge, our solution needs to include the projected ground-plane coordinates of the trajectories of pedestrians and it will be evaluated using a metric called Higher Order Tracking Accuracy (HOTA) score instead of an IDF1 score, which is the fraction of correctly identified detections over the average number of true and computed detections.

We aim to outperform the state-of-the-art solution, the 1st place winner of the 2023 AI City Challenge. We will compare the accuracies of each solution using the HOTA score of the results derived from the dataset given by the 2024 AI City Challenge.

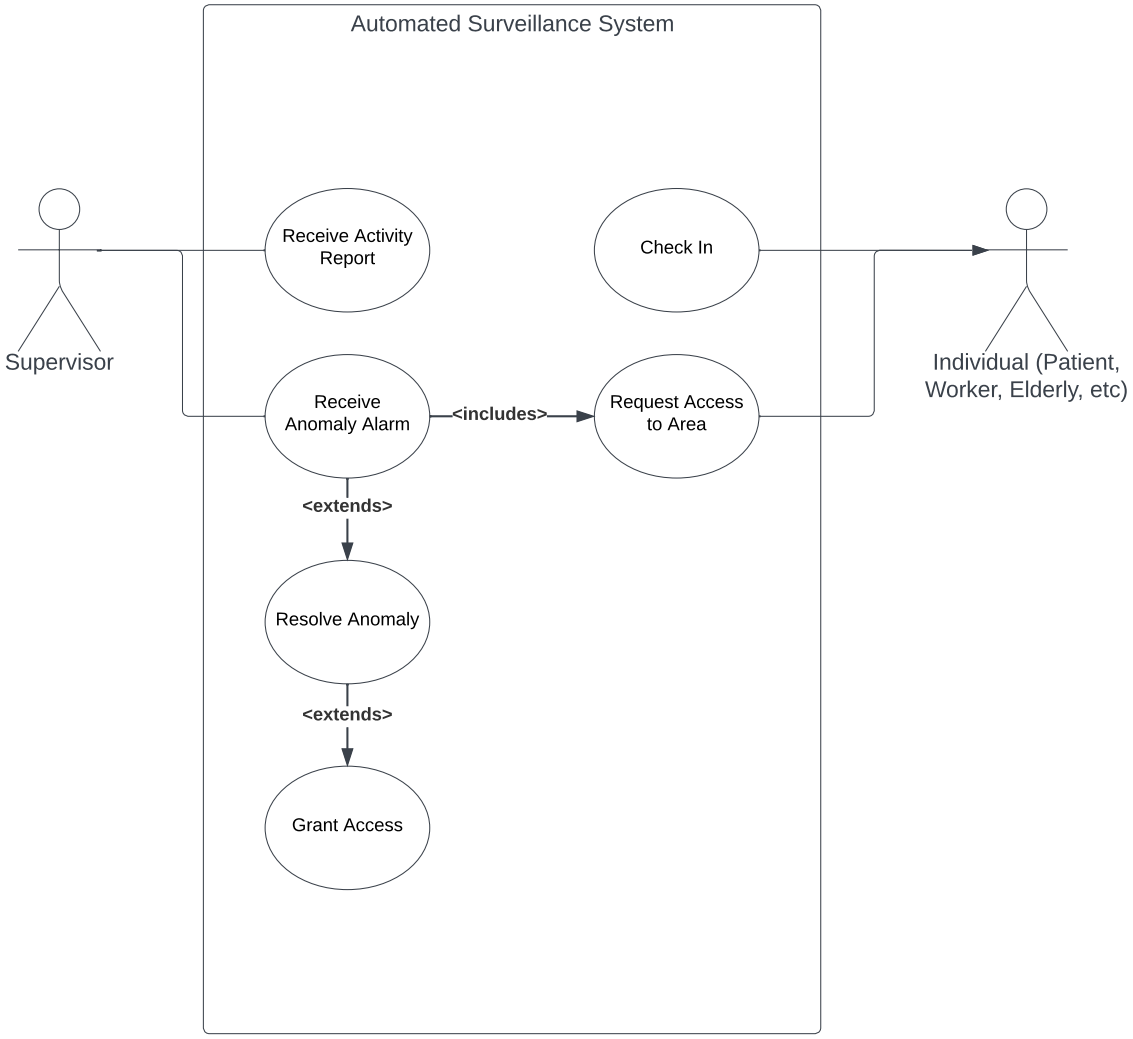

Figure 2.1: Use Case Diagram for an Automated Surveillance System

Our solution will not be deployed as an application. It serves to advance research regarding MCPT algorithms. As a result, there will be no functional requirements for usability or security. However, we have included a diagram in Figure 5.1 for a possible use case of MCPT, an Automated Surveillance System, with possible functional requirements.

2.2 Non-Functional Requirements

Although not a primary evaluation criterion of the AI City Challenge, the challenge encourages its competitors to develop online MCPT solutions. Online algorithms run in real-time by generating results as they observe each frame compared to offline algorithms that generate results after all of the video frames have been collected. Online algorithms

are largely much less accurate than offline algorithms because they must make inferences based on knowledge only up to the current moment in time. Online algorithms are encouraged for MCPT, however, because tracking pedestrians in real-time is the most possibly efficient way to automate MCPT without human intervention.

Chapter 3

Literature Survey

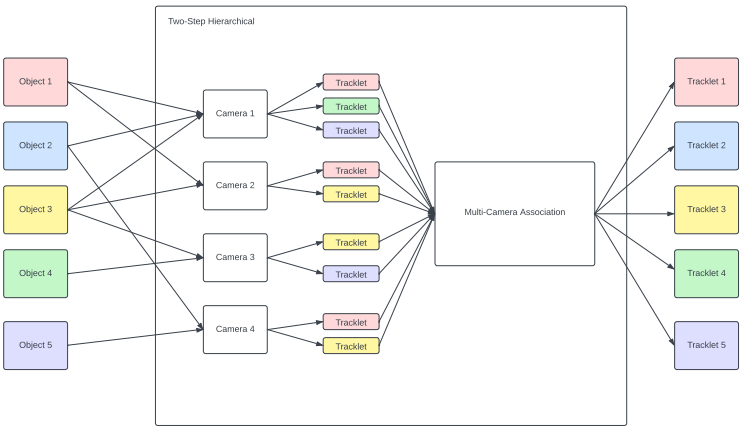

Figure 3.1: Two-Step Hierarchical Approach to Multi-Camera People Tracking

Two popular approaches exist for Multi-Camera People Tracking (MCPT): the two-step hierarchical approach and the global MCPT approach. The global MCPT approach tries to solve the problem by processing camera footage across the entire network at once using a single data association technique to assign global IDs to each pedestrian. In contrast, the two-step hierarchical approach addresses the problem by generating *tracklets*[1] for pedestrians within a single camera using Single Camera Tracking (SCT) module, then associating the single-camera tracklets across multiple cameras into trajectories using an Inter-Camera Association (ICA) module to assign a global ID to each pedestrian, thus using separate data association techniques for SCT and ICA. The two-step hierarchical approach assumes that the SCT module gives reliable results for MCPT and it is employed by the winning teams from the MCPT track in AI City Challenge 2023. Generally, a two-step hierarchical approach for MCPT involves 1) object

[1] short sequences of frames of a pedestrian with a consistent ID

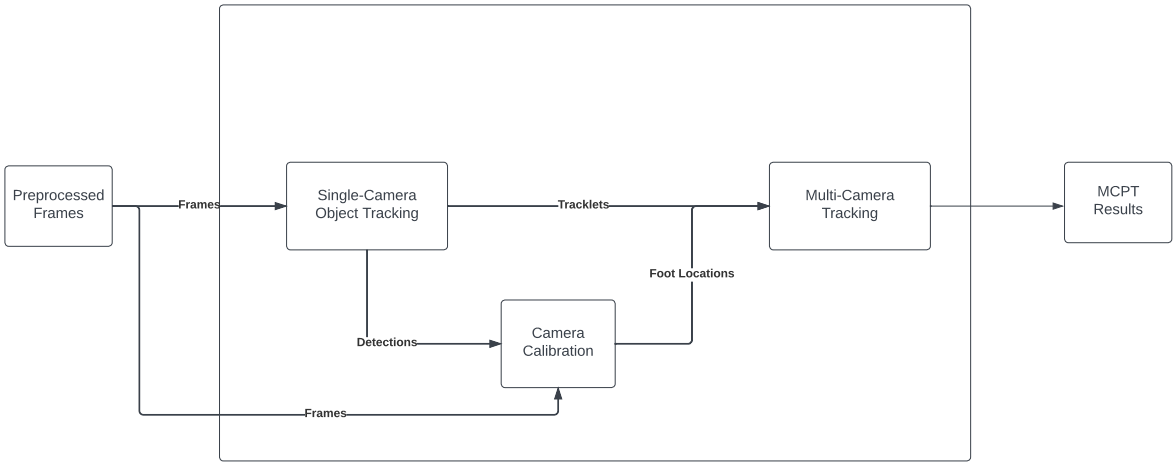

Figure 3.2: Multi-Camera People Tracking Architecture

detection, 2) a re-identification feature extraction model, 3) an SCT module to associate intra-camera tracklets, a projection to the 3D global space, and 4) an ICA module to assign global IDs in the 3D global space.

3.1 Object Detection

Object detection is an essential and initial step in solving MCPT, which employs systems to detect specific objects. The state-of-the-art model for object detection is the You Only Look Once (YOLO) model which detects human objects and defines their bounding boxes. Over the years, YOLO has undergone significant advancements, showcasing substantial improvements in detection performance and optimization. It is a mature and reliable tool in the realm of object detection.

3.1.1 YOLO

YOLO is a single-shot object detection method that uses a fully Convolutional Neural Network (CNN) to process images. An image is broken down into an S x S grid, and each cell predicts bounding boxes and their confidence scores. Once a grid cell detects the center of the object, the grid cell is responsible for detecting the object. Lastly, YOLO models use non-maximum suppression (NMS) during the post-processing step to improve accuracy by removing incorrect and redundant bounding boxes.

3.2 Re-Identification Feature Extraction

Re-identification (Re-ID) is the computer vision task of identifying pedestrians in a video over time. It's a critical component for solving SCPT and MCPT. In SCPT, re-ID discriminates between pedestrians with highly similar ap-

pearances and identifies pedestrians as they change poses and become occluded by objects or other pedestrians. In MCPT, re-ID identifies pedestrians as they travel across the field of vision of several non-overlapping cameras. It faces the same challenges as single-camera tracking as well as issues arising from variations in lighting conditions, camera angles, and video resolutions across cameras.

In the context of MCPT, re-ID is performed on videos using primarily deep learning methods to extract appearance features of pedestrians across several frames and aggregate spatio-temporal information. These features are fed to an SCPT module that identifies pedestrians according to the similarity of their features to generate single-camera tracklets, which are finally given to an MCPT module to track pedestrians across several cameras. Consequentially, information-rich re-ID features and their representation are the foundation for any effective MCPT solution.

Re-ID features are attributes of pedestrians, such as spatial, temporal, and appearance information that are used to identify and discriminate pedestrians in MCPT. In scenarios with high-accuracy requirements such as MCPT, it's preferred to extract semantic visual features using deep neural networks [5]. Several image-level feature extractors use deep neural networks to yield different feature representations of pedestrians. There are generally four categories for representing re-ID features: global, local, auxiliary, and sequence.

3.2.1 Global Feature Representation

The image-level feature extractor learns a holistic representation of each pedestrian from the entire image. Global features are simple to calculate but they are sensitive to noisy images and misaligned poses across frames[6]. Convolutional neural networks (CNNs) were used for image classification initially, so they were a popular approach for early re-ID solutions. Deep CNNs (DCCNs) have many more network layers to learn more semantic information, thus increasing accuracy. However, after a certain number of layers, accuracy stagnates and then decreases rapidly. Now, residual networks are widely preferred, namely ResNet, a network that uses convolutional layers yet achieves greater accuracy with more layers by using residual blocks, all while being computationally cheaper than a DCNN [7]. Kim et al performed MCPT re-ID by averaging features produced by three ResNet models: ResNet50-IBN, ResNet101-IBN, and ResNeSt-50 [8]

3.2.2 Local Feature Representation

Features are extracted from certain regions of the images and then the feature representation of pedestrians is a fusion of several local features. For instance, local features can represent certain body parts to improve robustness against misalignment of poses [9]. DCNNs and ResNet are used for extracting both global and local feature representations. OSNet is another popular DCNN-based network that uses residual blocks, but it learns richer discriminative features by using a combination of local and global features across several scales[10]. The winner of the 2023 AI City Challenge used OSNet to re-ID pedestrians in MCPT, choosing it for its ability to effectively fuse appearance features from

multiple scales [4]. Attention mechanisms such as self-attention can dynamically find and focus on relevant local features given a sequence of input images[11]. Transformers are a self-attention mechanism that can learn the importance of features within images efficiently in parallel[12]. TransReID is a model that uses a transformer to capture both global information and more discriminative parts of an image, outperforming CNNs which suffer from information loss and inability to focus on globally relevant patterns[13] Nguyen et. al. achieved state-of-the-art results for MCPT by employing an ensemble method for re-ID that combines the re-ID features extracted from both transformer-based and CNN-based models, namely TransReID, HRNetw48, and TransReID with Jigsaw Patch Module (JPM) and Side Information Embeddings (SIE) [14]. Li et al found through ablative studies that TransReID-based models performed substantially better than ResNet50-based models [15].

3.2.3 Auxiliary Feature Representation

Additional information is generated or extracted from the images such as semantic attributes to give a more comprehensive re-ID feature presentation. Generative Adversarial Networks (GANs) are deep neural networks that can generate new data to give re-ID solutions more diverse data. For example, GANs can fabricate images of a pedestrian in different poses to increase robustness to pose variation [16].

3.2.4 Sequence-based Feature Representation

While other feature representations extract features on a per-frame basis and then create the final feature either through average pooling or max pooling, in a sequence-based, or video-based, feature representation, each pedestrian's features are a fusion of features of each sequenced frame in a video, enriching the representation of pedestrians using appearance, spatial, and temporal information [17]. Some common challenges with sequenced-based feature representation include difficulties capturing temporal information, identifying pedestrians from sequences that are outliers, and comparing sequences of varying length [17]. Eom, et al created a Spatio-Temporal Memory Network (STMN) that remembers frequent spatial distractors and temporal patterns by levering a key-value structure and an LSTM to provide context for frame-level re-ID[18]. Zhang et al defined a new intra-video loss and combined it with a Siamese loss to better associate frames to videos with the same pedestrian while sufficiently discriminating between sequences of different pedestrians [19]. A sequence-based feature representation is often better for modeling spatio-temporal information except in cases of small datasets or poor video quality [5].

3.3 Single Camera Tracking

The Two-Step Hierarchical Approach naturally assumed a reliable Single Camera Tracking (SCT) module, and that a global homography matrix as well as clustering algorithm at a global scale is applied to the output of the SCT.

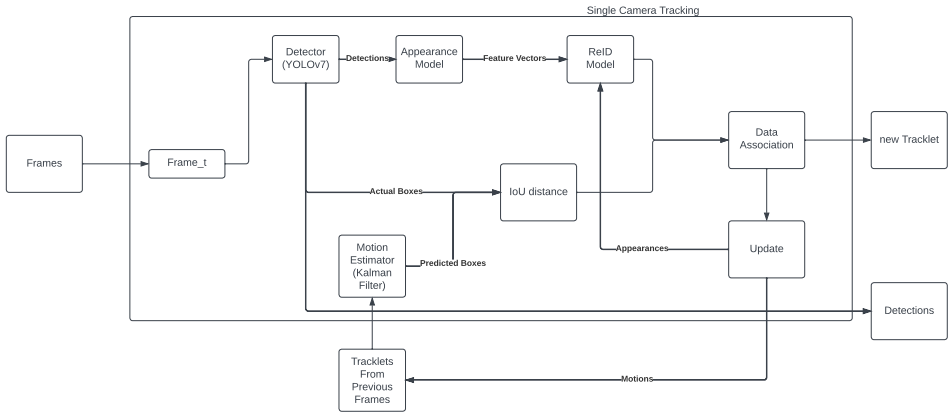

Figure 3.3: Single Camera Tracking Module

However, it is worth noticing that the state-of-the-art SCT models were still not robust enough and may experience ID switch in the context of occlusion.

3.3.1 Tracking by Detection

Many state-of-the-art models for multiple object tracking (MOT) tasks employ the tracking by detection scheme as proposed by Bewley et. al. in SORT [20] which involves separation of the detection and tracking tasks. A separate detector such as YOLOx proposed by Ge et. al. [21] is usually included to produce boxes around the target, and an association algorithm is applied to assign a unique ID. Such strategies rely on the detection outputs and have difficulties by nature handling occlusion because part of the target's body can be unavailable resulting in low confidence detection or displaced boxes. Online single-camera MOT algorithms also suffer from performance costs, which can seriously hinder the application of such algorithms.

3.3.2 Data Association

While the detection model is meant to produce anonymous targets only, extra work must be done to associate each detection box with an ID or other detections from previous frames. The most common approaches to this are using a re-identification (ReID) model, which compares the feature of detected targets, or using a motion model, often Kalman Filter, or both to acquire affinities between detected targets in adjacent frames. ByteTrack by Zhang et. al. [22], Bot-SORT by Aharon et. al., [23], SparseTRACK by Liu et. al. [24], SMILETrack by Wang et. al. [25] addressed the problem of occlusion by employing a cascade matching strategy where not occluded detections are matched first, and the occuluded or detections with lower confidence are matched with the unmatched IDs. SparseTrack noted that objects closer to the camera are less likely to be occluded and proposed using the pseudo-depth (simulated distance to the camera) of the object for cascade matching. Xu et. al. in their work TransCenter [26] addressed the problem

of occlusion by representing each detection using a center point instead of a box, which alleviates the difficulties of matching occluded detections due to the occluded part of the body.

From the AI City challenge 2023, Nguyen et. al. [27] and Yang et. al. [28] both attempted to alleviate the problem of ID-switch in the context of single camera tracking. Nguyen et. al. checked the distribution of appearance features from a single track to ensure the robustness of the algorithm, while Yang et. al. used the ReID model to compare the identified track to all other tracks that appeared in the same frame at the same time.

3.3.3 Motion Modeling

Many state-of-the-art MOT models such as ByteTrack, and BotSORT incorporate motion modeling of targets in their design. ByteTrack used Kalman Filter to predict the boxes of existing targets in the current frame from the motion model. The Intersection-over-Union (IoU) distance between predicted and detected boxes is then used in data association. BotSORT also proposed an improvement in the Kalman Filter, finding that directly predicting the width and height of a box as well as its derivatives can yield better results.

3.3.4 Feature Modeling

Models like BOT-SORT-ReID and SORT rely highly on the ReID models to perform data association. Normally these models will involve a pretrained ReID model for the association tasks. Some model like SMILETrack and TransCenter also uses self-attention mechanisms or convoluted neural networks (CNN) to learn a feature representation. Then cosine similarity is often used on the learned feature embeddings to generate an affinity matrix used in the data association process.

The Feature Pyramid Network (FPN) proposed by Lin et. al. [29] is also frequently employed to learn a deep feature representation. The FPN lowers the memory and time required compared to Featurized Image Pyramid learning, which is to learn a feature representation at different scales, and the FPN avoids losing semantic information compared to directly learning the feature representation from one scale.

While most projects rely on visual cues and spatial cues to calculate the similarity between tracklets, [30] proposed adding another layer that calculates and analyzes pose similarities in addition to a visual similarity layer and a spatial similarity layer. An additional layer of pose similarity can further improve the process of handling occlusion and fast motion. It is employed by Cao et. al. in OpenPose [31] to predict the key points for each proposal and employs Object keypoint Similarity (OKS) to compute pose similarities between targets.

Qiang et al. [32] also proposed correlation learning to represent an object using learned local temporal-spatial correlation with its neighbors. By incorporating the social information into the feature model, the model becomes more robust as the ID switch problem is remedied.

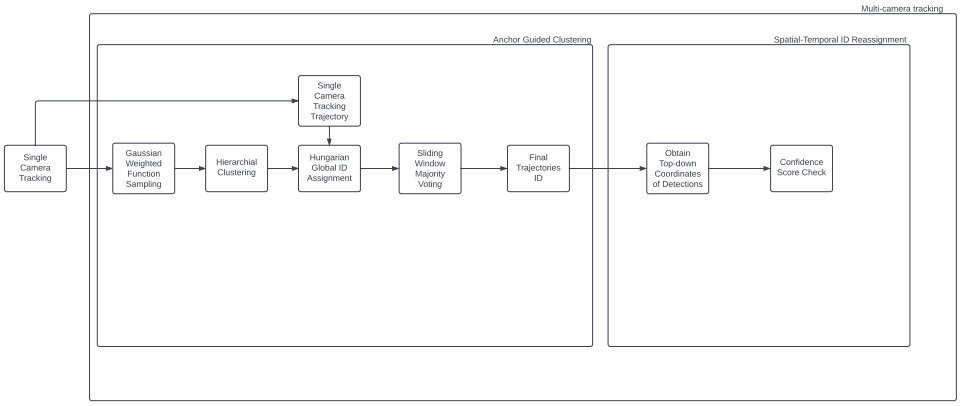

Figure 3.4: Multi Camera Module

3.4 Inter-Camera Association

Inter-Camera Association(ICA) refers to the process of associating tracklet information generated from Single-Camera Tracking to accurately track objects as they move across different camera views. Over the years, significant challenges have been found in the areas of global ID reassignment and ID switches because appearance features can be captured differently across camera views and object occlusions. 3.3 displays the proposed singular camera module by the winning team from AI City Challenge 2023

3.4.1 Data Association - Appearance Model

Numerous approaches have been proposed over the years to correctly identify observations across multiple camera views. They all commonly agree that there exists a strong indication for co-identity between two observations if there is a great similarity between their appearance features and spatial-temporal information.

One common approach in ICA to solve issues regarding ID switching is to perform clustering on the extracted appearance information and leverage spatial-temporal information to construct a spatial-temporal distance matrix to gain more accurate positions. The third place of the AI City Challenge, Young et al. [33], uses hierarchical clustering on appearance data to associate tracklets, and they leverage spatial-temporal constraints and integrate them into a distance matrix. Compared to methods that solely rely on clustering of appearance features, this method reflects better accuracy, as it can associate trajectories from different cameras better. More specifically, the spatial-temporal distance matrix incorporates intra-camera spatial information (homograph distance) and inter-camera spatial-temporal information (ID antinomy, speed constraint). The homography distance matrix is determined by identifying multiple matching points between camera views, ensuring these points are on the ground plane and meet specific geometric

criteria. ID antinomy refers to the phenomenon that multiple people in the same frame are assigned duplicate identities. To mitigate ID antinomy, it is penalized with a loss function. Speed constraint is also a factor that they considered when eliminating ID switching. Since there is a limit for human speed, two tracklets from the same camera cannot lead to the same person if the interval between one's beginning time is too short for the distance.

To resolve the difficulty of ID switches due to occlusion and target re-entry problem, Huang et al. [4] proposed a new clustering method, Anchor-Guided clustering, to target this problem. The method begins by periodically sampling the appearance features for a certain number of frames from each camera in the same scene. After that, hierarchical clustering is performed to obtain anchors for each identity, which contains features that represent each identity's appearance feature under different detection sizes, lighting conditions, and rotation angles. Each anchor will be assigned a unique ID, which represents the identity's global ID in multi-camera tracking. Once the anchors are obtained, the Hungarian algorithm will be performed with the following cost function. Lastly, after performing the Hungarian algorithm, each trajectory obtains a global ID list with the same length as the original trajectory. To obtain the final global ID, a sliding window majority voting approach is implemented, which effectively fixes ID switches in single-camera tracking.

The tracklet association problem can also be solved by treating it as a hierarchical structure optimization problem with two stages, designed by Xu et al. [34]. First, the hierarchical structure begins with the organization of the scene into a compositional hierarchy denoted as G, which is defined as G = (VN, VT, S, X).(VT = set of terminal nodes, VN = set of non-terminal nodes, S = the root node representing the entire scene, X = represents the set of states of both terminal and non-terminal nodes) Within the hierarchical structure, each tracklet contains appearance and geometry information over a certain period, which includes information on the 2D bounding box and 3D ground position and time stamps. Second, a Bayesian Formulation algorithm is applied to solve the problem of inferring the hierarchical composition.

The issue can also be addressed by reassigning the identity of tracklets to their nearest cluster by re-calculating the distance between each tracklet and the center of the cluster after conducting clustering on appearance features [35].

3.4.2 Probabilistic Occupancy

To improve the accuracy of determining the position of the trajectories, Fleuret et al. [36] a model with a probabilistic occupancy map can be employed by composing the probability of a target standing on each grid to identify trajectories. The model relies on a motion model, an appearance model, and a color model to retrieve enough information for its intention.

3.4.3 Tracklet-to-Target Assignment

Unlike many of the approaches that use the tracklet-to-tracklet matching approach, He et al. [37] proposed the tracklet-to-target matching method (TRACTA) solves tracklets matching by targeting two main issues with tracklet-to-tracklet approaches. First, the number of local tracklets across multiple camera views might be different, since different targets appear in a different number of cameras. Second, it is difficult to adhere to the *matching consistency principle*. In TRACTA, each tracklet is assigned to a unique target, and the optimal assignment is computed using the Restricted Non-negative Matrix Factorization (RNMF) algorithm. More specifically, the method can be broken down into four modules, 1)local tracklet generation module, 2)tracklet similarity measurement module, 3) cross-camera tracklet matching module, and 4) global trajectory generation module. The RNMF algorithm calculates the optimal tracklet-TID assignment matrix A* and can correct the tracking errors caused by occlusions and missed object detection in the previous stage.

3.4.4 ID Reassignment

Oftentimes, a global ID may be incorrectly assigned after using a data association technique due to highly similar appearances or occlusions. Therefore, to further improve the accuracy of global ID assignment, numerous methods are proposed to reassign global ideas after the data association is applied.

Spatio-Temporal Consistency ID Reassignment is a method [4] that corrects the incorrect global ID assignments by utilizing spatial-temporal information, which is made possible by calibrating the camera footage such that it projects the foot coordinates on a top-down view map. The process of camera calibration can project the ground plane for different camera views. The foot coordinates give a physical location for a pedestrian and may reveal any spatio-temporal inconsistencies. The 2D pose estimator, HigherHRNet, plays a part in determining the confidence scores of the key points.

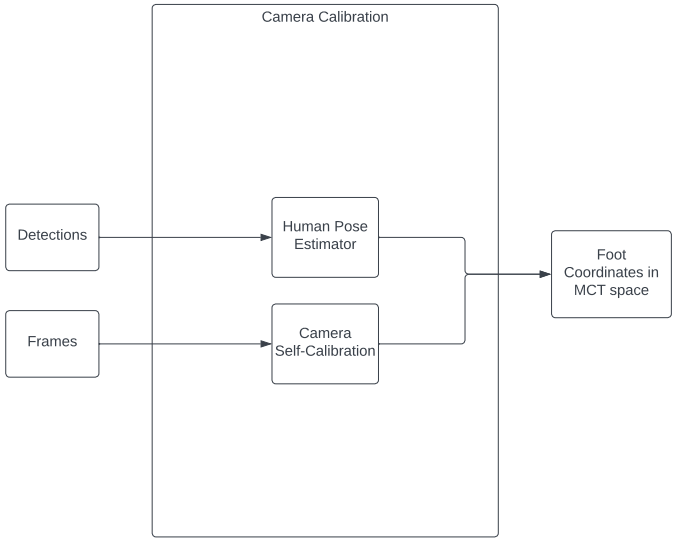

Figure 3.5: Camera Calibration Module

Chapter 4

Standards and Constraints

4.1 Standards Used

Because the use of deep learning techniques in computer vision is a relatively new practice, there exist few well-defined industry standards. Computer vision researchers have largely shaped the standards used through trial and error based on interoperability and quality of results. To ensure interoperability and obtain quality results, we use widely used open-source computer vision libraries. Many of these computer vision libraries were developed by researchers who designed and published their novel algorithms. We write our implementation using Python and use the Python version of these libraries because of efficiency, extensibility, and interoperability. The Python libraries we used during the implementation/testing phases of our project include PyTorch, NumPy, Scikit-Learn, OpenCV-Python, etc. We use the OpenCV-Python library to process frames in our dataset, and it is regarded as an industry standard for embedded vision [38]. The Python version of these libraries can execute complex algorithms in a few concise lines while running the actual computation efficiently using C++ in the backend. This allows us to compose different algorithms without worrying about memory management and strict types. Because Python is the industry standard programming language for computer vision, there exist many algorithms already implemented in Python, and many developers in the field are well accustomed to it, making our implementation easy to extend and compose with other algorithms if it is written in Python [39].

Our implementation works on the .mp4 input video format, which is the format of every video in the AI City Challenge dataset, and the intermediate results are committed to the file system using either plain text files or binary serialization of NumPy Objects (.npy format). All computation is performed on SCU's WAVE High-Performance Computing (HPC) Center using accelerators from NVIDIA to stay in compliance with the AI City Challenge's data license agreement. We also fall under any user limit raised by SLURM, the resource management system used in our HPC. The collaboration between multiple programmers was powered by Linux access control.

Furthermore, we modularize our pipeline so that future groups can modify or extend our pipeline. In Table 4.1, we list the technologies used by each module. In the Methods chapter, we detail the implementation of the modules

and technologies used.

4.1.1 ISO/IEC 14496 (MPEG-4)

The ISO/IEC 14496 (MPEG-4) [40] is the standard for the underlying technology for the .mp4 extension video encoding format. This was adopted explicitly by employing open-source software such as OpenCV which handles video format under this standard.

4.1.2 ISO/IEC 27001

The ISO/IEC 27001 [41] is the standard for information security mangement system, which enforces security practices and mitigates to a certain degree the ethical concerns that relates to data-security in the task of Multiple Camera People Tracking. This standard should be established with application of our proposed pipeline.

Module	Type of Technology	Technology
Detection	Single-shot detector	YOLOv8
Single Camera Tracking	Motion-based offline single camera tracker	UCMCTrack
Feature Extraction	Frame processing library	OpenCV
	Re-identification feature extractor	FastReID
Inter-Camera Association	Clustering algorithm	Anchor-guided clustering
	Camera calibration	Spatio-Temporal Consistency ID Re-assignment

Table 4.1: Technologies Used by Pipeline Module

4.2 Design Constraints

The efficacy of Computer Vision tasks like Multi-Camera People Tracking (MCPT) is greatly affected by the dataset's quality for training, validation, and testing. The underlying neural networks of MCPT solutions learn the observed patterns in the training dataset, and their performance is evaluated on the validation dataset and testing dataset. Moreover, training and executing these neural networks to make inferences takes plenty of time and computing resources. There are several constraints to consider when designing our MCPT solution regarding dataset availability and computing resources.

4.2.1 Dataset Availability

Because we are competing in the AI City Challenge, we must use their datasets to test our MCPT solution to ensure that the submitted solutions are evaluated fairly against each other. Although we began our Senior Design Project in October, we were not given access to the datasets until January 22, 2024. As a result, we had to design our solution without being able to properly test it for several months. Moreover, we had less time to test our solution because our submission is due on March 25, 2024. The dataset is different from the challenges from previous years; there are more videos to process and the videos have a higher resolution. Because there is more data for our MCPT solution to

process, it will take a longer computation time than participants in challenges from previous years. The given training dataset was also very limited. The scenes in the training dataset captured only a handful of challenging scenarios. There were not many hard cases of identifying highly similar-appearing pedestrians. In addition, the datasets were entirely of simulated environments with synthetic pedestrians in the NVIDIA omniverse, so our MCPT solution can only track synthetic pedestrians. Finally, we had to keep the datasets confidential. We were only allowed to store the datasets at Santa Clara University's WAVE High Performance Computing (HPC) Center. As a result, we could not move the datasets to run computations on another machine, which limited us to the shortcomings of the WAVE HPC Center.

4.2.2 Santa Clara University's WAVE High Performance Computing (HPC) Center

The WAVE HPC is a facility meant to serve the needs of faculty research and both undergraduate and graduate student research such as computing resources, large data storage, and advanced visualization. Upon request, students and faculty can be given access to the WAVE HPC. While only a subset of SCU's students and faculty have access, the WAVE HPC is a shared resource with a finite number of compute nodes, storage capacity, and memory. As a result, there is almost always competition for resources. To run a job on one of the compute nodes, wait times range from several minutes to several hours. Additionally, there are only four GPU nodes in the entire facility, making these nodes a scarce resource. This can be problematic because many tasks such as training neural networks are unfeasible on anything other than a GPU. Lastly, each job has a time limit that it can run until it is terminated by the WAVE HPC. All of the GPU nodes have a time limit of two days, which is oftentimes not enough time for the job to complete its execution.

Another constraint that has to be considered is the availability of the file system. Since the file system is shared, we receive a disk quota on a per-user basis, making it unrealistic to perform frame extraction as employed by most existing object-detection and Re-ID frameworks. To overcome this, our solution is implemented in a streaming fashion, eliminating the need to extract frames, but is very unfriendly to the detector training/fine-tuning tasks as they lack randomization.

Chapter 5

Design Methods

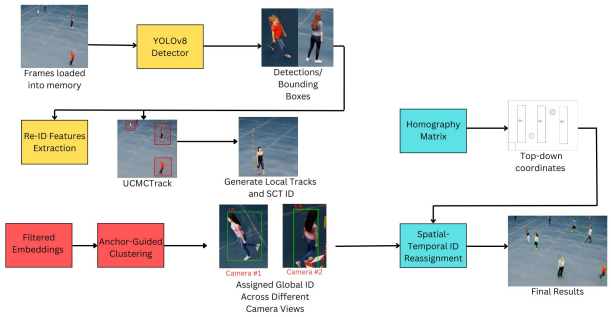

Figure 5.1: Two-Step Hierarchical Multi-Camera People Tracking Pipeline

5.1 Single Camera Tracking

Single-camera tracking algorithm is used after the detections of targets are generated using the YOLOv8 detector, including bounding boxes on the targets, which are necessary components of of the tracking process. We employed UCMCTrack [42] as our single-camera tracking algorithm to generate local trajectories, and it offers a few advantages over other popular tracking algorithms, such as BotSORT and Deep-OC-Sort. Unlike most of the other single-camera tracking algorithms, UCMCTrack operates solely on motion cues, which elevates the efficiency of the model without sacrificing the accuracy of the tracking results.

Although our algorithm is evaluated on a synthetic dataset, UCMCTrack has the potential to handle real-life scenarios that will inevitably contain camera movement. UCMCTrack handles this problem by adding random noise to the homography matrix.

5.2 Feature Extraction

Extracting the appearance features of the targets is an essential step before the Inter-Camera Association and clustering. We used FastReID[43], developed by the JDAI research team, a PyTorch toolbox for real-world person re-identification. For the baseline model for deep person re-identification, we chose to use Bag-of-Tricks (BOT) with ResNET50 as the backbone network, to accurately extract appearance-based embeddings from the detections that are generated from YOLOv8 detector across different cameras. Furthermore, to elevate storage efficiency, we loaded frames into memory using the CV2 library, rather than extracting and storing the frames locally.

5.3 Inter-Camera Association

Inter-camera association aims at comparing and mapping anonymous tracklets with others across cameras to generate coherent tracks that will globally and uniquely identify a human object. To accomplish this, we used appearance features from the videos to re-identify every tracklet by checking their similarity. This idea is, however, obscured due to many tracking and re-id challenges, such as object occlusion which would hinder the precision of re-ID matching; different viewing angles that introduced variability to the re-ID feature for even the same object, and imbalanced re-id feature distribution for people who stay in the scenes longer versus those who don't stay. Our solution pipeline includes filtering of the re-ID feature banks and clustering algorithm applied to the filtered features and an optional reassignment algorithm that handles conflicts as they arise.

5.3.1 Variance Preserving Filtering

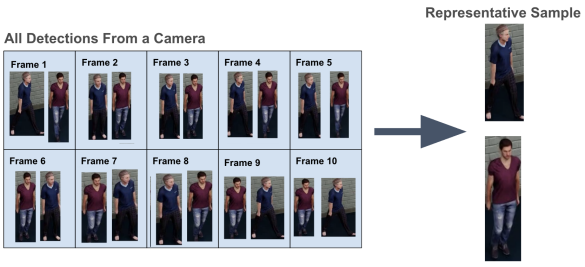

Figure 5.2: Sampling the detections across all cameras to obtain the best representative samples

Our algorithm relies on obtaining accurate and comprehensive feature clusters for each of the uniquely identifiable individuals who appeared in the scenario. If clustering is conducted without pre-processing, the computational power needed for clustering will surge rapidly beyond the capacity of any modern computer. Previous works [4] considered sampling frames periodically to reduce the number of features, but such an approach wouldn't be robust for some scenarios where people are only briefly observed by the camera, and where a sampling rate must be manually balanced on a per-scenario basis to find a trade-off between robustness and efficiency. The Variance-Preserving Filter-

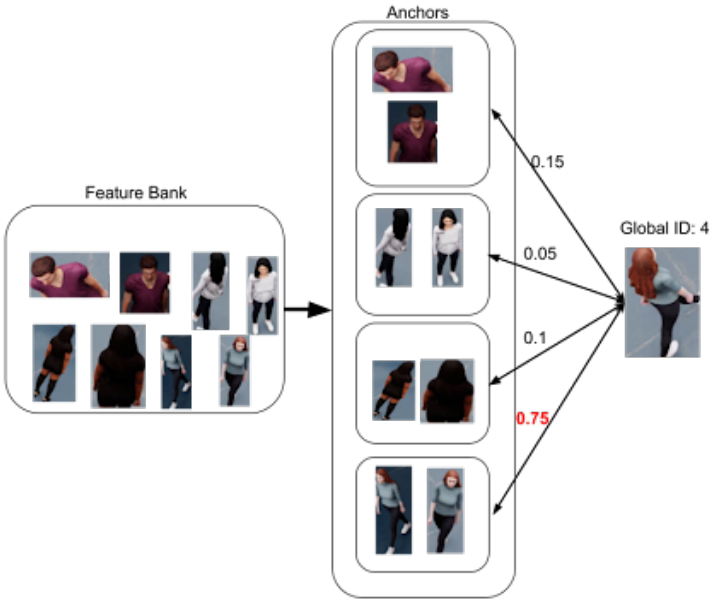

Figure 5.3: Obtaining Anchors from Bank of Features

ing algorithm aims to effectively obtain only the most useful information about every individual by utilizing a Python script to compare the cosine similarities between appearance-based embedding. To achieve this, we would remove features from the occluded objects, as they would interfere with the clustering performance; and we would reject any features that are redundant to features we have seen previously, as described by Figure 5.2. This algorithm allowed us to increase our clustering quality at a much lower cost.

5.3.2 Anchor Guided Clustering

The quality of the feature cluster obtained for each individual is crucial to our design. The agglomerative clustering algorithm is the most suited for our design because in our case we do not know beforehand how many people have entered the scene. The clustering algorithm is performed on the highly varied samples coming from the filter and in light of the re-ID models each cluster will only contain features from a uniquely identifiable individual under different cameras as described in Figure 5.3, different frames, and different locations. Such a cluster is called an Anchor and will give us enough information about the identity of a tracklet when compared to its Re-ID feature. Hungarian algorithm is then used to associate tracklets, but instead of comparing tracklets to tracklets directly, we find the similarity between tracklets to anchors instead. This mechanism is retrospective and will help resolve the re-entering problem where one object can be occluded for a short period before it appears in the frame again, causing a sudden change in appearance.

5.3.3 Spatio-Temporal Consistency ID Reassignment

In reality, perfect clustering and association of tracklets hardly happen because faulty results in earlier stages of the pipeline (e.g. missed/extra detections, ineffective features, bad tracklets) can alter the correctness of the inter-camera association. Most of the issues will result in some kind of conflict, for example, if the same global ID is associated in two different locations in the same camera view, etc. the Spatio-Temporal Consistency ID Reassignment will detect these conflicts and reassign them to the next closest or a new global ID as appropriate.

Chapter 6

Evaluation

6.1 Evaluation Methodology

We evaluated the performance of our algorithm using the Higher Order Tracking Accuracy (HOTA) [44] score, which is a new metric that is used to measure the performance of multi-object tracking, such as multi-camera people tracking (MCPT). HOTA incorporates three key aspects of tracking performance: detection accuracy, association accuracy, and localization accuracy. Detection accuracy measures how well the tracker detects objects in each frame, association accuracy measures how well the tracker assigns and maintains the correct ID of objects across frames, while localization Accuracy measures how accurately the tracker estimates the position and size of the objects in each frame. Compared to other existing metrics like IDF1 and MOTA, HOTA provides a more stable and reliable evaluation.

6.2 Accuracy and Performance Comparison Against Baseline

The proposed solution is executed and compared to the accuracy and performance of the state-of-the-art solution that placed 1st in the 2023 AI City Challenge, which we regard as the baseline solution [4]. We arbitrarily chose scene 044 from the validation dataset, provided by the AI City Challenge for comparison. Table 5.1 presents the detection accuracy (DetA), association accuracy (AssA), localization accuracy (LocA), and higher order tracking accuracy (HOTA) metrics for the baseline and proposed solutions on scene 044. The proposed solution achieves comparable DetA (53.9% vs 59.5% baseline) and slightly improved LocA (91.6% vs 89.4%). However, the proposed method significantly outperforms the baseline on AssA (15.9% vs 21.4%) while having moderately lower HOTA (30.5% vs 35.6%).

Breaking down the execution time by stages in Table 5.2 reveals where the efficiency gains are made. The proposed solution eliminates the 1 hour 11 minutes spent on frame extraction in the baseline solution. We reduced the time to generate detections from the frames from 2 hours 31 minutes to just 3 minutes 4 seconds. Re-ID feature extraction sees a huge reduction in execution time from over 15 hours to just 47 minutes. We also improve the execution time for clustering by reducing it from 3 hours 43 minutes to 17 minutes. The baseline spends over 15 hours

	Baseline	Our Solution
DetA	59.5	53.9
AssA	21.4	15.9
LocA	89.4	91.6
HOTA	35.6	30.5

Table 6.1: Score breakdown for the results of Baseline solution and the proposed solution on Scene 044

Execution Time by Stages		
Stages	Baseline	Our Solution
Frame Extraction	1 Hour 11 Minutes	N/A
Generate Detections	2 Hours 31 Minutes	3 Hours 4 Minutes
Re-ID Feature Extractions	15 Hours 29 Minutes	47 Minutes
Single Camera Tracking	14 Minutes	2 Minutes
Clustering	3 Hours 43 Minutes	17 Minutes
Spatial-Temporal ID Reassignment	15 Hours 40 Minutes	3 Minutes
Total	38 Hours 48 Minutes	4 Hours 13 Minutes

Table 6.2: Execution time breakdown by stages for Baseline solution and the proposed solution on Scene 044

on spatial-temporal ID reassignment, while the proposed method needs only 3 minutes for this stage. In total, the optimized approach condenses the pipeline from 38 hours 48 minutes to 4 hours 13 minutes, an 89% reduction.

6.3 Results

Our method achieves a HOTA score of 44.72% tested in 100% of the AI City Challenge 2024 dataset, which consists of 30 scenarios with 12 videos in each scenario. Picture 6.1 shows our result in the official leaderboard. This score was attained from a submission that was a few days after the official challenge deadline but would have placed us in the 8th place if submitted before the deadline.

Rank	Team ID	Team Name	Score
1	221	RIIPS	71.9446
2	79	SJTU-Lenovo	67.2175
3	40	NetsPresso	60.9261
4	142	FraunhoferIOSB	60.8792
5	8	UWIPL-ETRI	57.1445
6	50	ARV RETERIU	51.0556
7	5	SKKU-AutoLab	45.1575
8	1	SCU_Anastasiu	44.7285
9	124	STCHD	40.6202
10	162	Asilla	40.3361

Figure 6.1: AIC 2024 Leaderboard

Chapter 7

Societal Issues

7.1 Ethical Justification

Use cases for multi-camera people tracking (MCPT) systems monitor the movement of pedestrians to automate processes that either improve security or efficiency. Some examples include retail theft detection and warehouse worker route optimization. Utilitarianism argues that the most moral course of action is one that brings the most good for the most amount of people. From a utilitarian lens, MCPT systems are justified because improving security using an MCPT system will ensure safety for most people, and optimizing processes will maximize profits. However, utilitarianism treats individuals as a means to an end, and many people can still be negatively affected, so there is a need to design MCPT systems using other ethical frameworks.

Several ethical frameworks can ensure that MCPT systems respect the individuals they are monitoring. A rights-based approach argues that individuals have certain moral rights that must be protected and respected under any circumstances. Data privacy is a critical ethical consideration from a rights-based approach to designing MCPT systems. From a justice and fairness ethical framework, bias, and discrimination must be prevented to guarantee that all individuals that are being monitored are treated fairly. Careful measures must be taken to not incorrectly identify one person for another based on similar appearance. Finally, from a care ethics lens, we must be wary of the environmental impact of the computationally intensive neural networks that underlie the MCPT system. In the following section, we discuss how we address these ethical challenges in our solution and the remaining pitfalls.

7.2 Ethical Challenges

7.2.1 Bias and Discrimination

MCPT systems, like many other computer vision technologies, can struggle with bias and discrimination. Take for instance an MCPT system that detects retail theft. The software may have difficulty distinguishing people of similar race, age, gender, etc due to variations in conditions across cameras including lighting, camera angles, and video quality. Therefore, if a person of a particular race is determined to be a shoplifter by the MCPT system, then another

person of a similar race may be mistaken to be the shoplifter. To prevent MCPT systems from incorrectly identifying similar-looking people, we must collect sufficient data for the neural networks of MCPT systems to make accurate decisions. The neural networks learn patterns of appearance by "training" on footage of pedestrians. The footage for each type of demographic should be a representative sample of the population of people it will be monitoring, and the sample size should be large enough to recognize general patterns. In our design, we use pre-trained neural networks to detect and extract attributes to distinguish similar-looking individuals. We chose to use pre-trained neural networks because of time constraints from Senior Design deadlines and computational resource constraints of Santa Clara University's HPC. The person detection neural network was already trained on Microsoft's Common Objects in Context (COCO) dataset. The dataset includes images of several classes including real persons. However, this dataset is highly biased. It has "more than twice the number of images of men than women", making it difficult for the MCPT system to correctly detect women [45]. Moreover, the COCO dataset had approximately "9.2x more images of lighter-skinned than darker-skinned people", so darker-skinned people may not be properly detected [45]. In contrast, the feature extraction neural network was trained on a synthetic dataset where the footage is of digital persons in a simulated environment generated by NVIDIA Omniverse. While these synthetic environments can be used to simulate many scenarios, digital persons don't perfectly imitate real persons in the field of view of a camera. As a result, our feature extraction neural networks cannot sufficiently extract attributes from individuals to accurately distinguish them from similar-looking individuals.

7.2.2 Data Privacy

A lot of data is required to sufficiently train the underlying neural networks of MCPT systems. Collecting the video footage to train these neural networks can violate data privacy. Oftentimes, neural networks are trained on data that has been scraped from the internet. The individuals in these videos were not informed about how their data would be ultimately used. There is a need for informed consent for training data collection. The images in Microsoft's Common Objects in Context (COCO) dataset that was used to train our MCPT system's person detection neural network were collected from Flickr, a website where amateur photographers can upload pictures [46]. It is unclear if the people in these pictures were aware that their images would be used to train many detection neural networks. The dataset used to train our MCPT system's feature extraction neural network was entirely of synthetic digital persons, so there was no informed consent to be collected. Because it can be difficult to obtain the informed consent of individuals for every potential use case that their footage is used for, it is appealing to use synthetic images of pedestrians. However, there exists a hit to accuracy when using training on images of synthetic pedestrians to track real pedestrians because the neural networks are learning the patterns of synthetic pedestrians to make an inference about real pedestrians.

There are data privacy concerns when deploying MCPT systems as well. For instance, when using MCPT to optimize the routes of workers in a warehouse scenario, or surveilling customers in a retail store to detect theft,

the individuals being monitored must consent to their data being recorded and analyzed. These individuals must be provided with informed consent. Moreover, they must have a right to know their data, a right to opt out of having their data collected, and a right to delete their data if they choose.

7.2.3 Environmental Impact

Neural networks are computationally intensive, so they produce a lot of emissions. According to researchers from the University of Massachusetts, Amherst, training a single neural network can emit more than 626,000 pounds of carbon dioxide [47]. Moreover, large neural networks such as ChatGPT produce 3.82 metric tons of CO2 per day from its 10 million queries. The neural networks underlying MCPT systems are no exception. We ran the neural networks of our MCPT system on NVIDIA V100 Tensor Core GPUs which each consume 500 watts of power at maximum [48]. In comparison, it takes approximately 1200 watts to power a home throughout the day [49]. Our MCPT system is designed to have a lessened environmental footprint than the state-of-the-art MCPT system because we use pre-trained neural networks and reduce the amount of footage to process by filtering out redundant data. When running both of our solutions using the same hardware, we achieve a much faster execution time than the state-of-the-art MCPT system.

Chapter 8

Conclusion and Future Work

8.1 Conclusion

8.1.1 Lessons Learned

By completing this project, we were able to make ourselves familiar with the HPC and the SLURM, a Linux utility for resource management, as we needed to use them repeatedly to run our programs. In addition, we sharpened our skills in coding in Python and interacting with some of the important packages, such as Numpy, Scikit-learn, Pandas, and SciPy.

This task is particularly challenging due to its memory-and-computation-heavy nature, which forced us to emphasize the resource limitations in our design and implementation process. We are punished for inefficient design or implementation. To bypass the file system and memory limitation, we configured our detection to process the data in a streaming fashion. To work with our algorithm efficiently, we parallelized our tasks into different SLURM tasks.

Managing environments and getting the accelerators to work is another thing we learned. Anaconda was used to comply to the varying versions of libraries used in different parts of the project. Getting PyTorch to work with our available GPU as well as deciding the most efficient batch-size was another challenge for us initially.

The most precious lesson we learnt from this project is that for enormous datasets like this, we should start building our project using a small subset of the dataset before we conduct a full-scale test. Realizing this facilitated our development process, which allowed us to make more submissions.

8.2 Future Work

Overall, our pipeline and algorithm can handle the problem of inaccurate tracking due to occlusions effectively. Still, several enhancements could be made in the future to improve the accuracy and performance of our pipeline. One improvement is to conduct better tuning on detectors and collect auxiliary information to help us further resolve the re-entry problem, especially when the targets are occluded for a long duration. Additionally, we can improve our pipeline's performance by investigating our algorithm's potential in an online scenario to allow tracking in real-time

across a network of cameras.

8.2.1 Detector Tuning

We can get decent results from our current detector, which was pre-trained using the COCO dataset. However, we can further improve the accuracy of our detector by fine-tuning it. For instance, since each camera view may vary in terms of lighting conditions, angle views, and backgrounds, fine-tuning the detector to adapt to each of the cameras can reduce false positives and false negatives, leading to better detection results.

8.2.2 Performance Improvement

Albeit the time efficiency achieved by our current algorithm, it is still, unfortunately, an offline algorithm, and has a problem scaling with the number of people. We should investigate the potential of our algorithm in an online scenario where tracking is done in real-time only using historical data.

8.2.3 Extension to Online Methods

While our current solution runs at an acceptable speed, it is unfortunately an offline method that utilizes information beyond the current time frame. However, we are confident that with minor modifications, our algorithm should be able to work with real-time data and qualify as an online method. Things to worry about include maintaining a feature bank with a manageable size, initial assignment to first-seen objects, etc.

8.2.4 Extension to Real-World Data

The video inputs provided by the AICity Challenge are generated via Nvidia Omniverse, meaning they are all synthetic. Real-world data from our experience behave differently from synthetic ones mostly in the sense that 1. Real subject's non-constant walking speed, 2. Noise induced by random camera motion, 3. Variety of human postures, and 4. social interactions. While our pipeline should work with real-world data out-of-the-box or with minor modifications, they might not achieve the same accuracy.

Bibliography

[1] A. Martin, "NVIDIA helps retail industry tackle its $100 billion shrink problem," Jan. 2023. Accessed: 2024-6-11.

[2] S. Mirza and S. Krishnamurthy, *Multi-Camera Image Tracking on a Global Plane*. 2021.

[3] Navigine, "Warehouse tracking systems."

[4] H.-W. Huang, C.-Y. Yang, Z. Jiang, P.-K. Kim, K. Lee, K. Kim, S. Ramkumar, C. Mullapudi, I.-S. Jang, C.-I. Huang, and J.-N. Hwang, "Enhancing multi-camera people tracking with anchor-guided clustering and spatio-temporal consistency id re-assignment," 2023.

[5] Z. Sun, X. Wang, Y. Zhang, Y. Song, J. Zhao, J. Xu, W. Yan, and C. Lv, "A comprehensive review of pedestrian re-identification based on deep learning," *Complex & Intelligent Systems*, pp. 1–36, 2023.

[6] D. Wu, H. Huang, Q. Zhao, S. Zhang, J. Qi, and J. Hu, "Overview of deep learning based pedestrian attribute recognition and re-identification," *Heliyon*, vol. 8, no. 12, p. e12086, 2022.

[7] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 770–778, 2016.

[8] J. Kim, W. Shin, H. Park, and J. Baek, "Addressing the occlusion problem in multi-camera people tracking with human pose estimation," in *2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, pp. 5463–5469, 2023.

[9] D. Yi, Z. Lei, S. Liao, and S. Z. Li, "Deep metric learning for person re-identification," in *2014 22nd International Conference on Pattern Recognition*, pp. 34–39, 2014.

[10] K. Zhou, Y. Yang, A. Cavallaro, and T. Xiang, "Omni-scale feature learning for person re-identification," in *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 3701–3711, 2019.

[11] M. Guo, T. Xu, and J. Liu, "Attention mechanisms in computer vision: A survey," in *Computational Visual Media*, vol. 8, p. 331–368, 2022.

[12] S. Khan, M. Naseer, M. Hayat, S. W. Zamir, F. S. Khan, and M. Shah, "Transformers in vision: A survey," *ACM Comput. Surv.*, vol. 54, sep 2022.

[13] S. He, H. Luo, P. Wang, F. Wang, H. Li, and W. Jiang, "Transreid: Transformer-based object re-identification," in *2021 IEEE/CVF International Conference on Computer Vision (ICCV)*, (Los Alamitos, CA, USA), pp. 14993–15002, IEEE Computer Society, oct 2021.

[14] Q. Q.-V. Nguyen, H. D.-A. Le, T. T.-T. Chau, D. T. Luu, N. M. Chung, and S. V.-U. Ha, "Multi-camera people tracking with mixture of realistic and synthetic knowledge," in *2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, pp. 5496–5506, 2023.

[15] Z. Li, R. Wang, H. Li, B. Wei, Y. Shi, H. Ling, J. Chen, B. Liu, Z. Li, and H. Zheng, "Hierarchical clustering and refinement for generalized multi-camera person tracking," in *2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, pp. 5520–5529, 2023.

[16] J. Liu, B. Ni, Y. Yan, P. Zhou, S. Cheng, and J. Hu, "Pose transferrable person re-identification," in *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 4099–4108, 2018.

[17] M. Ye, J. Shen, G. Lin, T. Xiang, L. Shao, and S. H. Hoi, "Deep learning for person re-identification: A survey and outlook," *IEEE Transactions on Pattern Analysis & Machine Intelligence*, vol. 44, pp. 2872–2893, jun 2022.

[18] C. Eom, G. Lee, J. Lee, and B. Ham, "Video-based person re-identification with spatial and temporal memory networks," 2021.

[19] W. Zhang, Y. Li, W. Lu, X. Xu, Z. Liu, and X. Ji, "Learning intra-video difference for person re-identification," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 29, no. 10, pp. 3028–3036, 2019.

[20] A. Bewley, Z. Ge, L. Ott, F. Ramos, and B. Upcroft, "Simple online and realtime tracking," in *2016 IEEE International Conference on Image Processing (ICIP)*, pp. 3464–3468, 2016.

[21] Z. Ge, S. Liu, F. Wang, Z. Li, and J. Sun, "YOLOX: exceeding YOLO series in 2021," *CoRR*, vol. abs/2107.08430, 2021.

[22] Y. Zhang, P. Sun, Y. Jiang, D. Yu, F. Weng, Z. Yuan, P. Luo, W. Liu, and X. Wang, "ByteTrack: Multi-object Tracking by Associating Every Detection Box," in *Computer Vision – ECCV 2022* (S. Avidan, G. Brostow, M. Cissé, G. M. Farinella, and T. Hassner, eds.), (Cham), pp. 1–21, Springer Nature Switzerland, 2022.

[23] N. Aharon, R. Orfaig, and B.-Z. Bobrovsky, "Bot-sort: Robust associations multi-pedestrian tracking," *ArXiv*, vol. abs/2206.14651, 2022.

[24] Z. Liu, X. Wang, C. Wang, W. Liu, and X. Bai, "SparseTrack: Multi-Object Tracking by Performing Scene Decomposition based on Pseudo-Depth," 2023. _eprint: 2306.05238.

[25] Y.-H. Wang, J.-W. Hsieh, P.-Y. Chen, M.-C. Chang, H. H. So, and X. Li, "SMILEtrack: SiMIlarity LEarning for Occlusion-Aware Multiple Object Tracking," 2023. _eprint: 2211.08824.

[26] Y. Xu, Y. Ban, G. Delorme, C. Gan, D. Rus, and X. Alameda-Pineda, "TransCenter: Transformers With Dense Representations for Multiple-Object Tracking," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 45, no. 6, pp. 7820–7835, 2023.

[27] Q. Q.-V. Nguyen, H. D.-A. Le, T. T.-T. Chau, D. T. Luu, N. M. Chung, and S. V.-U. Ha, "Multi-camera people tracking with mixture of realistic and synthetic knowledge," in *2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, pp. 5496–5506.

[28] W. Yang, Z. Xie, Y. Wang, Y. Zhang, X. Ma, and B. Hao, "Integrating appearance and spatial-temporal information for multi-camera people tracking," in *2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, pp. 5260–5269.

[29] T.-Y. Lin, P. Dollar, R. Girshick, K. He, B. Hariharan, and S. Belongie, "Feature Pyramid Networks for Object Detection," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, July 2017.

[30] S. You, H. Yao, and C. Xu, "Multi-target multi-camera tracking with optical-based pose association," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 31, no. 8, pp. 3105–3117, 2021.

[31] Z. Cao, T. Simon, S.-E. Wei, and Y. Sheikh, "Realtime multi-person 2d pose estimation using part affinity fields," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 7291–7299, 2017.

[32] Q. Wang, Y. Zheng, P. Pan, and Y. Xu, "Multiple Object Tracking With Correlation Learning," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 3876–3886, June 2021.

[33] W. Yang, Z. Xie, Y. Wang, Y. Zhang, X. Ma, and B. Hao, "Integrating appearance and spatial-temporal information for multi-camera people tracking," in *2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, pp. 5260–5269, 2023.

[34] Y. Xu, X. Liu, Y. Liu, and S.-C. Zhu, "Multi-view people tracking via hierarchical trajectory composition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 4256–4265, 2016.

[35] A. Specker and J. Beyerer, "Reidtrack: Reid-only multi-target multi-camera tracking," in *2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, pp. 5442–5452, 2023.

[36] F. Fleuret, J. Berclaz, R. Lengagne, and P. Fua, "Multicamera people tracking with a probabilistic occupancy map," *IEEE transactions on pattern analysis and machine intelligence*, vol. 30, no. 2, pp. 267–282, 2007.

[37] Y. He, X. Wei, X. Hong, W. Shi, and Y. Gong, "Multi-target multi-camera tracking by tracklet-to-target assignment," *IEEE Transactions on Image Processing*, vol. 29, pp. 5191–5205, 2020.

[38] A. Nandanwar, "An industrial overview of open standards for embedded vision and inferencing."

[39] Apr 2024.

[40] 14:00-17:00, "ISO/IEC 14496-33:2019."

[41] 14:00-17:00, "ISO/IEC 27001:2022."

[42] K. Yi, K. Luo, X. Luo, J. Huang, H. Wu, R. Hu, and W. Hao, "UCMCTrack: Multi-Object Tracking with Uniform Camera Motion Compensation," *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 38, pp. 6702–6710, Mar. 2024.

[43] L. He, X. Liao, W. Liu, X. Liu, P. Cheng, and T. Mei, "Fastreid: A pytorch toolbox for general instance re-identification," *arXiv preprint arXiv:2006.02631*, 2020.

[44] J. Luiten, A. Ošep, P. Dendorfer, P. Torr, A. Geiger, L. Leal-Taixé, and B. Leibe, "HOTA: A Higher Order Metric for Evaluating Multi-object Tracking," *International Journal of Computer Vision*, vol. 129, no. 2, pp. 548–578, 2021.

[45] Y. Hirota, Y. Nakashima, and N. Garcia, "Gender and racial bias in visual question answering datasets," in *2022 ACM Conference on Fairness, Accountability, and Transparency*, FAccT '22, ACM, June 2022.

[46] T.-Y. Lin, M. Maire, S. Belongie, L. Bourdev, R. Girshick, J. Hays, P. Perona, D. Ramanan, C. L. Zitnick, and P. Dollár, "Microsoft coco: Common objects in context," 2015.

[47] E. Strubell, A. Ganesh, and A. McCallum, "Energy and policy considerations for deep learning in nlp," 2019.

[48] "NVIDIA V100 — NVIDIA — nvidia.com." [Accessed 18-05-2024].

[49] "How Many Watts Does it Take to Run a House? — energysage.com." [Accessed 18-05-2024].