REAL-TIME JOINT MULTI-CAMERA MULTI-PERSON TRACKING

A THESIS SUBMITTED TO
THE GRADUATE SCHOOL OF NATURAL AND APPLIED SCIENCES
OF
MIDDLE EAST TECHNICAL UNIVERSITY

BY

ABDUSSAMET TARIK TEMÜR

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS
FOR
THE DEGREE OF MASTER OF SCIENCE
IN
COMPUTER ENGINEERING

MAY 2024

Approval of the thesis:

REAL-TIME JOINT MULTI-CAMERA MULTI-PERSON TRACKING

submitted by **ABDUSSAMET TARIK TEMÜR** in partial fulfillment of the requirements for the degree of **Master of Science in Computer Engineering Department, Middle East Technical University** by,

Prof. Dr. Naci Emre Altun
Dean, Graduate School of **Natural and Applied Sciences** _____

Prof. Dr. Halit Oğuztüzün
Head of Department, **Computer Engineering** _____

Assoc. Prof. Dr. Emre Akbaş
Supervisor, **Computer Engineering, METU** _____

Examining Committee Members:

Prof. Dr. Sinan Kalkan
Computer Engineering, METU _____

Assoc. Prof. Dr. Emre Akbaş
Computer Engineering, METU _____

Assist. Prof. Dr. Cemil Zalluhoğlu
Computer Engineering, Hacettepe University _____

Date:22.04.2024

I hereby declare that all information in this document has been obtained and presented in accordance with academic rules and ethical conduct. I also declare that, as required by these rules and conduct, I have fully cited and referenced all material and results that are not original to this work.

Name, Surname: Abdussamet Tarık Temür

Signature :

ABSTRACT

REAL-TIME JOINT MULTI-CAMERA MULTI-PERSON TRACKING

Temür, Abdussamet Tarık
M.S., Department of Computer Engineering
Supervisor: Assoc. Prof. Dr. Emre Akbaş

May 2024, 49 pages

This study aims to construct a Real-Time Multi-Camera Multi-Person Tracking (MCMOT) system which jointly optimizes local (single-camera) and global (multi-camera) feature distances. While most existing approaches follow a two-stage track-then-associate scheme, this work focuses on a joint approach. Our method also operates in real-time in contrast to the more common offline or windowed joint tracking algorithms which operate on future information. In summary, this study contributes: (i) A joint MCMOT formulation where the optimization objective solves both local and global tracking at teach step, (ii) a realization of the method in the form of an algorithm capable of producing real-time track IDs, and (iii) a new MCMOT evaluation metric we call Global IDF1 which acts as a multi-camera extension of the IDF1 metric, emphasizing continuous traceability of a target across a multi-camera network. We further propose a Multi-View Fusion (MVF) network to extract descriptive feature vectors for multi-camera detection groups. We report results comparable to offline state-of-the-art methods while remaining real-time and retaining simplicity.

Keywords: Tracking, Multi-Camera, Real-Time, Multi-Object, Re-Id

ÖZ

GERÇEK ZAMANLI BÜTÜNLEŞİK ÇOKLU KAMERA ÇOKLU İNSAN TAKİBİ

Temür, Abdussamet Tarık
Yüksek Lisans, Bilgisayar Mühendisliği Bölümü
Tez Yöneticisi: Doç. Dr. Emre Akbaş

Mayıs 2024 , 49 sayfa

Bu çalışma, yerel (tek kamera) ve global (çoklu kamera) özellik mesafelerini birlikte optimize eden Gerçek Zamanlı bir Çoklu-Kamera Çoklu-Kişi Takip (MCMOT) sistemi oluşturmayı amaçlamaktadır. Mevcut yaklaşımların çoğu iki aşamalı, önce tek kamerada takip yapıp sonra takipleri bağlayan bir şema izlerken, bu çalışma ortak bir yaklaşıma odaklanmaktadır. Yöntemimiz, daha yaygın olan ve gelecekteki kareler üzerinde çalışan çevrimdışı takip algoritmalarının aksine gerçek zamanlı olarak sonuç üretir. Özetle çalışmamızın katkısı şu şekildedir: (i) Optimizasyon hedefinin her adımda hem yerel hem de global benzerlikleri dikkate aldığı bir ortak MCMOT formülasyonu, (ii) formülasyonun, gerçek zamanlı takip yapan bir algoritma ile gerçekleştirilmesi, ve (iii) IDF1 metriğinin çoklu kamera uzantısı olan ve bir çoklu kamera ağı içerisinde hareket eden hedeflerin sürekli izlenebilirliğini ölçen yeni bir MCMOT değerlendirme metriği olan Global IDF1. Ayrıca, takip edilen hedeflerin farklı perspektiflerden gelen görüntülerini bir arada temsil eden özellik vektörleri çıkarmak üzere özgün bir Çoklu Görüş Birleştirme (MVF) ağı önermekteyiz. Yöntemimiz, sadeliğini kaybetmeden gerçek zamanlı bir şekilde en iyi çevrimdışı yöntemlerle

karşılaştırılabilir sonuçlar üretmektedir.

Anahtar Kelimeler: İnsan-Takibi, Takip, Çoklu-Kamera, Çoklu-Nesne, Gerçek-Zamanlı

To family and friends.

ACKNOWLEDGMENTS

First and foremost, I would like to extend my gratitude to my thesis advisor Assoc. Prof. Dr. Emre Akbaş without whom this work would not have been possible. I would like to thank him for his support and guidance in my academic journey and dating back to my formative undergraduate years. I feel truly fortunate for having had the pleasure of working with him academically and professionally in the foundational years of my career.

I would like to express my heartfelt gratitude to Dr. Aykut Dengi and Assoc. Prof. Dr. Heni Ben Amor for their inputs and guidance. Their deep understanding of everything computer science has helped shape my conception of the entire field and my many learnings from our work together has undoubtedly carried over to this thesis.

I would like to thank my long time friend and colleague Ekrem Odabaş for his support and for the many invaluable discussions we shared over the course of my work. I have counted on his feedback and intuition on many occasions during the course of my work and elsewhere.

I would like to thank my work colleagues with whom we have explored the field of computer vision. Our teamwork has had a significant influence over my understanding of the field and its contributions to this thesis are incalculable. My special thanks goes to my former supervisor Stephen McCracken for our many discussions and conversations over the years. His endless supply of knowledge and experience lit the way where it was dark. And to my colleague and friend Orhun Köse whose peerless work ethic and methodology influenced me greatly.

Last but not least, I would like to express my eternal gratitude to my family for their love and support throughout my journey. I felt relieved, knowing they had my back regardless of outcome, and motivated, thanks to their endless and often contagious drive. This work was only possible thanks to their unwavering support and unconditional love.

TABLE OF CONTENTS

LIST OF TABLES

TABLES

LIST OF FIGURES

FIGURES

LIST OF ALGORITHMS

ALGORITHMS

LIST OF ABBREVIATIONS

ABBREVIATIONS

2D	2 Dimensional
3D	3 Dimensional
FPS	Frames per Second
GT	Ground Truth
HRI	Human-Robot Interaction
IoU	Intersection over Union
LSA	Linear Sum Assignment
MAP	Mean Average Precision
MCMOT	Multi-Camera Multi-Object Tracking
MCMOT	Multi-Camera Multi-Object Tracking
MCT	Multi-Camera Tracking
MOT	Multi-Object Tracking
MVF	Multi-View Fusion
NN	Neural Network
OCP	Optimal Camera Placement
SCT	Single-Camera Tracking

CHAPTER 1

INTRODUCTION

1.1 Motivation and Problem Definition

Multi-camera multi-object tracking (MCMOT) is a central problem in computer vision that involves the tracking of multiple objects across multiple cameras simultaneously. In contrast to the comparatively more simple classical (single-camera) tracking problem, multi-camera multi-object tracking requires that targets be tracked across several cameras with varying degrees of overlap at different viewpoints. In light of current developments in fields such as self-driving vehicles [6, 7, 8], smart cities [9], human-robot interaction [10, 11, 12, 13] and many others, the importance of jointly solving the tracking problem across large numbers of cameras and sensors has significantly increased.

Posing MCMOT as a joint optimization problem where single-camera and multi-camera features and distances are both taken into account is a desirable goal pursued by many previous studies [14, 15, 16, 17, 18]. The dominant paradigm for joint MCMOT uses various graph formulations of the problem, thus requiring time-window based or offline operation. In this study, we propose a real-time method capable of jointly optimizing over single-camera and multi-camera feature distances in a joint optimization objective. Real-time solutions are particularly valuable for applications like human-robot interaction and smart vehicles, where quick decision-making is essential and sometimes a hard requirement.

1.2 Proposed Methods and Models

We propose a Real-Time Multi-Camera Multi-Person Tracker using visual and geometric features. We make use of 2D ground plane projections of detection targets as geometric, multi-camera or global features which requires a calibrated camera network. Our algorithm fuses these feature distances and jointly optimizes over the final score using the Hungarian algorithm [19] to assign each detection a global track ID in a single stage approach at each frame. Figure 1.1 shows a general overview of our method, for a calibrated multi-camera network, we extract detections and visual/geometric features at each time step, then compute track IDs.

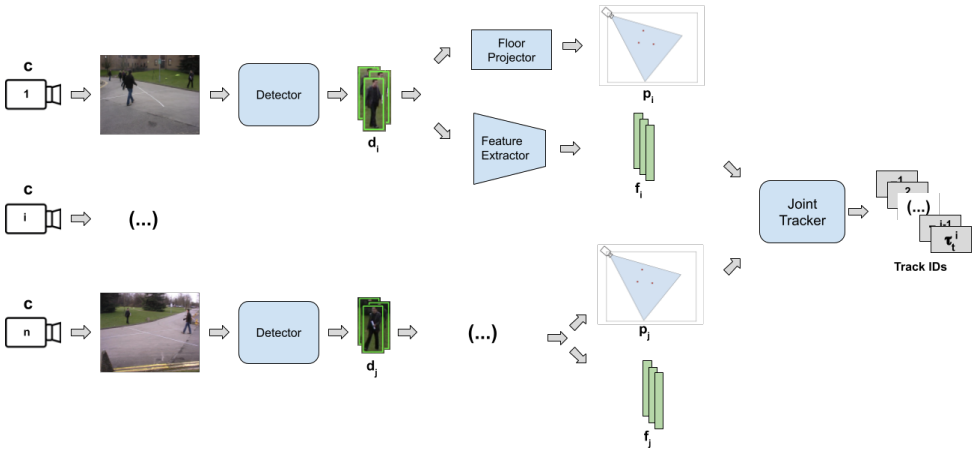

Figure 1.1: Overview of our approach. Given an n camera video stream, at each frame t, our tracking pipeline produces detection boxes for all cameras. Then, we extract floor projections and visual features for each detection before passing the information into the Joint Tracker module. The Joint Tracker produces globally consistent track IDs for all input detections.

The Joint Tracker module first produces cross-camera detection groups to act as person hypotheses, then matches these groups to existing global tracks based on a fused distance function using visual and floor projection features. We explain the Join Tracker in detail in Chapter 3.

1.3 Contributions and Novelties

Graph based formulations dominate the state-of-the-art in multi-camera multi-person tracking. Most graph based tracking algorithms must operate with a time window or offline for successful graph construction. We propose a joint optimization approach to real-time multi-camera multi-person tracking which optimizes multiple sources of feature distances jointly, producing track IDs for each detection at each frame in contrast to window-based or offline graph formulations. We also present a realization of this approach which uses visual-feature vectors extracted by a NN and ground-plane projections. We further fuse single camera visual features via a novel transformer based fusion network. We finally propose a new evaluation metric, Global IDF1, to better capture a global tracker's ability to correctly and continuously track a target across a multi-camera network.

1.4 The Outline of the Thesis

This thesis is structured in 5 chapters. The first chapter provides a general overview of the method and explains the contributions and novelties introduced by our work. The second chapter provides a detailed overview of existing work where we survey a large number of papers from the the multi-camera multi-person tracking literature as well as other related problems. We also provide a comprehensive analysis of the real world use cases for our research. The third chapter contains and explanation of the proposed method in detail using figures and pseudocode. We also provide an in depth explanation of the main assumptions we make in our study. The fourth chapter begins with a comprehensive survey of existing performance metrics for tracking, we then propose a new tracking metric and present a comparative analysis. Further, we provide quantitative results and benchmarks on public multi-camera multi-person tracking datasets and report the computational complexity and run times of our method. We finally present ablation studies where we remove certain components of our system and discuss the effects on performance. In the fifth and final chapter, we discuss the main limitations of our work and potential ways of addressing them, which leads to a discussion of future work.

CHAPTER 2

BACKGROUND AND RELATED WORK

This chapter provides an overview of the history and state-of-the-art in real-time multi-camera multi-object tracking (MCMOT) and other fields and sub-fields relevant to our study. We use persons (pedestrians) as tracking targets in our work, therefore, the background and relevant research explored in this section also focuses heavily on the person-tracking formulation of MCMOT.

We begin by providing a comprehensive, high-level background for the problem and exploring tracking in single and multi-camera settings with emphasis on real-time applications. We then explain ground plane projections in calibrated multi-camera networks, which are essential to our work. We finally conclude with a survey of related studies from the literature.

2.1 Background

Visual tracking has been a central problem in image processing and computer vision from its inception. The problem can be defined most generally as the tracking of an object or group of objects through time in a single or multi-camera network. Although visual tracking has a rich history, with influential studies dating back to the late 1990s [20], the multi-camera formulation of the problem has experienced a surge in research interest in recent years. This growing focus is partly driven by advancements in camera technology and availability, coupled with enhanced computational capabilities at the edge.

In visual tracking, the number of tracking targets and cameras vastly change the prob-

lem difficulty by introducing a plethora of new possible scenarios. This has lead to the field of visual tracking fragmenting into smaller sub-fields of research based on specific problem formulations. In this work, we focus on the most challenging formulation; multi-camera multi-object tracking.

Multi-camera multi-object tracking (MCMOT) is defined as the problem of tracking multiple objects across multiple cameras simultaneously. Although some ideas presented in our work can be generalized to different tracking targets, we focus on tracking persons in this study. With human tracking-targets, MCMOT is an increasingly more important problem in computer vision due to its various downstream applications in rapidly developing fields such as surveillance for security [21, 22, 23], self-driving vehicles [6, 8] human-robot interaction [10, 11, 12], smart cities [9], intelligent transportation [7] and others.

Many applications of MCMOT greatly benefit from real-time solutions, as they often enforce strict time windows within which trackers must operate. Human-robot interaction and smart vehicles are two great examples to these applications. In both problem definitions, the decision-making algorithm depends on one or several trackers that must operate within very short time windows, ruling out offline or large-window solutions. We further explore the importance and implications of developing real-time tracking solutions in this chapter.

Several sub-fields of research under visual tracking largely overlap with our problem definition. Studies in these fields provide rich context and powerful insights for our work. We mainly explore three adjacent problem definitions: Real-Time SCMOT, POI Tracking and Offline MCMOT.

Real-Time SCMOT (Single-Camera Multi-Object Tracking), sometimes referred to as simply Real-Time MOT (Multi-Object Tracking), is the problem of tracking multiple objects given a video stream from a single camera in real time. This domain contains a robust literature with many insightful studies for its multi-camera extension some of which are further explored in this chapter. POI (Person of Interest) tracking is the problem of tracking a specific person of interest across a multi-camera system in real-time. This is a very important problem for many security and law enforcement applications and is a relaxed formulation of Real-Time MCMOT. Finally offline MC-

MOT has the same formulation as our problem with the real-time constraint relaxed or removed. Figure 2.1 offers a conceptual view of the position of this study within relevant fields of research.

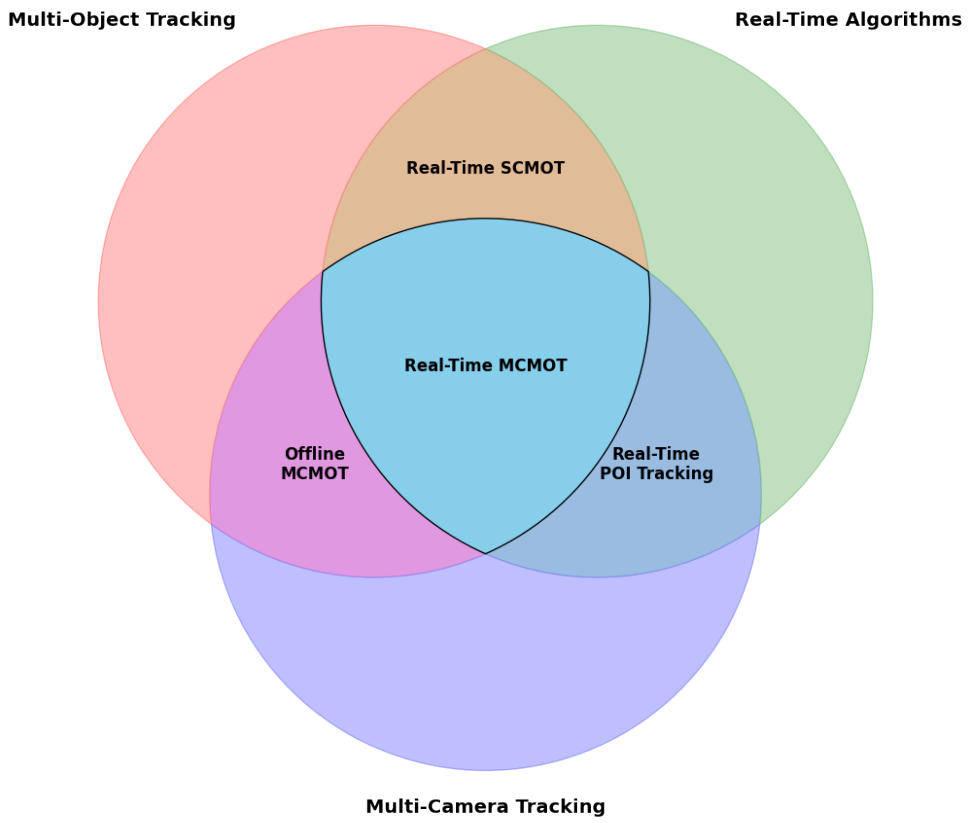

Figure 2.1: Venn diagram showing the position of our work within three central fields of research in visual tracking. We position our work at the intersection of MCMOT with real-time algorithms. Adjacent fields of research, namely; Real-Time SCMOT, Offline MCMOT and POI Tracking, are formulated by removing one of our three important constraints.

Also worthy of note is the mobility of the camera-network on which tracking is to be done. Algorithms for both SCMOT and MCMOT further branch into specializations for still-camera vs moving-camera networks. In our work, we mainly focus on the still-camera setting, however the central ideas can be extended to moving-camera systems.

2.1.1 Single-Camera Tracking

In the space of visual tracking, Single-camera tracking (SCT) is defined traditionally as the problem of tracking the movement of one or more objects across time given a continuous video stream. The multi-target version of the problem is termed single-camera multi-object tracking (SCMOT). Terminology around SCT frequently skips explicitly specifying the single-camera nature of the problem setting, resulting in SCT being referred to simply as 'Tracking', and SCMOT being referred to simply as 'MOT'. We, however, adopt a more precise nomenclature in this study.

Literature in SCT has a long history, with earlier approaches adapting methods from different fields to visual tracking for predominantly robotics applications. In their 1995 paper, Lee et al. apply Kalman filters to visually tracking a single object moving within a 3D volume given a video stream [20] with impressive results. Later studies expand on these methods by generalizing them to multi-object settings [24, 25] and addressing difficult scenarios.

There are various challenges in single camera tracking from handling variations in object appearance to dealing with partial or full occlusions and lighting differences. Another significant difficulty comes from low or variable frame rates and pacing. Especially for non-rigid and asymmetric objects like pedestrians, variations in visual appearance over time pose a significant problem for visual feature based trackers. Various studies attempt to alleviate these difficulties using a plethora of methods. Salscheider [26] fuses motion clues and visual features using an SVM based method to address visual variations while Chen et al. [27] use combinations of feature vectors from multiple different depths in a CNN based feature extractor. To account for illumination differences, Yang et al. [28] propose a Hyperline Clustering based method increasing the robustness of color histogram based trackers.

A large body of studies focus on alleviating visual variability difficulties for SCMOT. As most of these methods attempt to increase feature robustness, they are directly applicable to our Real-Time MCMOT solution as well as most other MCMOT solutions.

Finally, we define the predominant tracking-by-detection paradigm [29] which we

also adopt in our work. The tracking-by-detection paradigm is a three-stage paradigm where a continuous stream of frames are passed through a *detector*, producing *detections*. The *detections* are then processed by one or several *feature extractors*, producing *feature vectors*. These are then sent to the *tracker* which uses feature distances and various other specific logic to process the inputs and assigns *track ids* to each detection. Figure 2.2 provides an overview of the paradigm. All state-of-the-art and baseline methods we explore and use for comparison in this study follow the tracking-by-detection paradigm [18, 30, 31, 32, 33, 34, 17] unless otherwise specified, as does our own method.

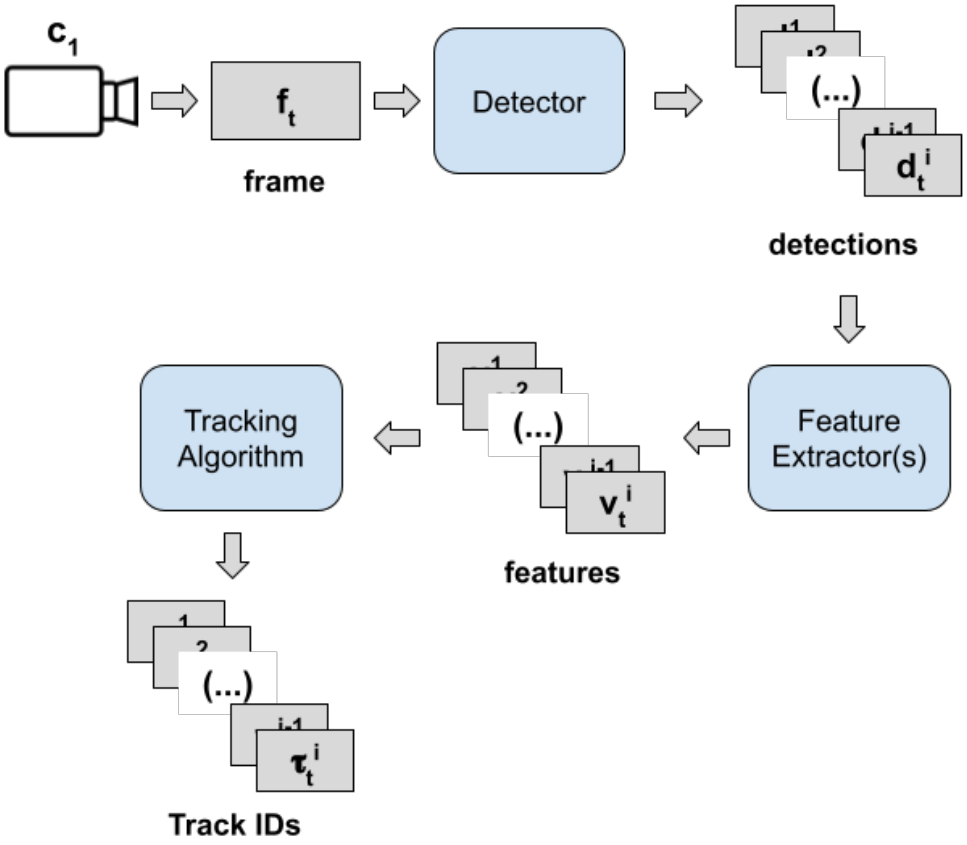

Figure 2.2: Overview of the tracking-by-detection paradigm. At frame *t*, a frame is sent from camera c_1 to the *Detector*, which then produces zero or more detections *d* and passes them to the feature extraction stage where features are extracted. The tracking algorithm then produces track IDs using feature distances.

There is also a detection-free tracking paradigm [35, 36, 37], where the goal is to track hand-picked (or otherwise specified) objects across a video. The main advantage of

detection-free tracking is its ability to track an object regardless of its class, given that most object detectors are trained on a specific class or set of classes. More recently, a Track-Anything system [38] was proposed, building a detection-free tracker on top of Meta's Segment-Anything [39] model. Although detection-free tracking is also an interesting field of research, we focus on the tracking-by-detection paradigm in this work.

2.1.2 Multi-Camera Tracking

Multi-Camera Tracking (MCT) is the extension of Single-Camera Tracking into multi-camera networks. In MCT, an additional goal of re-identifying persons across cameras is introduced, complicating the problem further. MCT further branches into single and multi-object formulations. The single object formulation is used to solve the Person-of-Interest (POI) Tracking problem, where the goal is to track a specific person across a multi-camera network. We focus on the Multi-Object formulation (MCMOT) of the problem and further set our tracking target objects as humans. Although MCMOT has implications for POI Tracking, it is outside of the scope of this study other than being a potential downstream application.

The dominant paradigm for MOT is *tracklet* association where single-camera trackers are ran on each camera's output stream, then the resulting single-camera tracks, called *tracklets*, are re-identified across cameras using a re-identification module [40, 18]. Figure 2.3 provides a high level overview of this paradigm. State-of-the-art in this paradigm is dominated by models attempting to fix tracklets at the multi-camera stage [41, 18], where the re-identification algorithm has the further capability of dividing input tracklets based on cross-camera information. Studies outside of this paradigm are predominantly formulations posing MOT as a graph problem where nodes are detections as opposed to tracklets [41].

As the re-identification layer is downstream of the single-camera trackers in this paradigm, the real-time nature of the algorithm is dependent on the upstream trackers. Most commonly, the re-identification algorithm only makes an assignment once it is given a full tracklet, which leads to variable length delays for the final global ID assignments [41, 42, 22]. That is, if a tracklet is t seconds long, it will only be assigned

its final global ID after t seconds. In our work, we assign global track IDs to each detection at each frame, resulting in a closer to real-time solution limited only by the camera's Frames-per-Second (FPS) parameter.

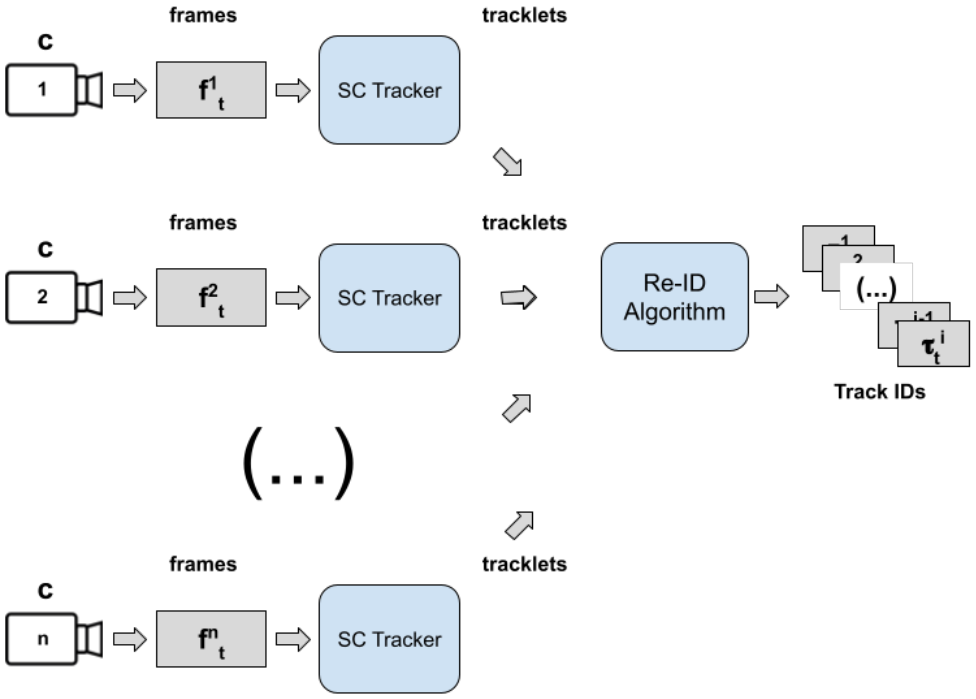

Figure 2.3: Overview of the tracklet association paradigm for Multi-Camera Tracking. In this paradigm, a single-camera tracker is ran on each video stream. The output tracklets and their relevant features are then passed onto a re-identification algorithm to be matched into final global track IDs.

2.1.3 Ground Plane Projections as Global Features

Ground plane projections represent footfall positions of person detections on a 2D coordinate system representing an approximately flat floor plane. In a multi-camera network, the cameras can be calibrated and registered into the same 2D floor coordinate system, making ground plane projections a unified multi-camera feature for MCMOT. Several studies attempt to make use of ground plane projections for MCMOT [18, 43, 44], with all reporting promising results. We also make use of ground plane projections as multi-camera features in this work.

Ground plane projections requires the camera-network to be calibrated, which may be a difficult process depending on the scene. This limits the availability of open source datasets and introduces representational variation in calibrations. In this study, we focus on the WILDTRACK [3], PETS09 [2] and EPFL (Terrace and Walkway) [1] datasets, all of which offer ground plane projection capabilities. We further discuss the datasets and calibration representations in the next chapter.

2.2 Related Work

This section details existing work in the field of tracking with an emphasis on studies adjacent to real-time and joint multi-camera multi-object tracking. We first analyze the existing work on Joint MCMOT, then move on to explore near or fully real-time solutions.

2.2.1 Joint MCMOT

We define Joint MCMOT as the task of jointly optimizing over single-camera and multi-camera feature distances to arrive at the final global track IDs. That is, given a detection from some camera in a multi-camera network, a joint tracker produces a track ID based on existing single-camera tracks and cross-camera tracklet associations jointly with neither taking full precedence.

In its original form, tracklet association shown in Figure 2.3 works by using a downstream re-id module whose objective is to connect tracklets. Potential ID switch errors in the tracklets caused by the upstream single-camera trackers are not accounted for in the re-identification module [40, 42].

Several studies have attempted to solve tracking as a global optimization problem. The most prominent approach to globally optimizing over local and global feature distances is by posing MCMOT as a graph problem where nodes represent detections or small groups of detections, and edges represent similarities between connected nodes. The similarity scores are calculated using feature distances.

Dehghan et al. [16] propose a Generalized Maximized Multi-Clique Problem for-

mulation to MCMOT where graph nodes represent small batches of detections and show promising results. Chen et al. [17] use a joint distance function taking both single-camera and multi-camera distances into account in a global graph formulation. Further, Nguyen et al. [18] bring a Deep Learning based approach to the formulation by proposing NN models capable of dividing single-camera tracks based on multi-camera information. This also allows the final model to be a true joint model.

Although graph based formulations have been used to produce joint tracking solutions [16, 17, 18], one shortcoming they have is that they must work either in a sliding window based fashion or completely offline. In contrast, we propose a joint MCMOT solution capable of working in real-time, producing track IDs for each detection at each frame.

2.2.2 Real-Time MCMOT

The ability to track and associate person detections across a multi-camera network is a desirable ability for many applications. In human-robot interaction (HRI), for instance, a very important problem is human intention recognition [13, 45, 46] defined as the problem of recognizing a human's intentions for the immediate future. Human intention recognition is a crucial problem for HRI as predicting human behavior is a must for successful interaction. Unsurprisingly, this use case requires real-time or close to real-time operation.

Real-time or near real-time multi-object tracking has a robust literature with contributions from various fields of application. You et al. [43] propose an end-to-end real time tracking pipeline called Deep Multi-Camera Tracking (DMCT) using customized neural networks for computing floor projections and track IDs at each frame. Gaikwad et al. [22] develop a fast deep learning based multi-camera tracker specialized Nvidia Jetson edge machines achieving up to 30 frames per second (FPS) with their re-identification algorithm. Yang et al. [47] propose an online distributed tracking system which makes use of visual feature vectors extracted by a neural network as well as histograms generated by image patches. There are many other studies focusing on Real-Time MCMOT.

CHAPTER 3

METHOD

We propose a real-time tracking solution to multi-camera multi-object tracking for person tracking which fuses single-camera and multi-camera features at the detection level to relate detections to tracks at each frame in a single-stage algorithm. Given a set of synchronized cameras overlooking a scene with acceptable coverage, our algorithm provides global track ids for each person detection at each frame. This chapter formally defines and rigorously explains the problem, with emphasis on its setting and underlying assumptions.

Our model works on a synchronized multi-camera network where the cameras are calibrated and both intrinsic and extrinsic calibration parameters are known. The cameras produce frames at each time-step which are then passed through a detector, producing zero or more person detections for each camera. The detections are then passed through the feature extraction stage where we (i) compute the bounding box center coordinates, (ii) pass the bounding box crops through a feature extraction network which consists of a Body-part-Based feature extraction network [4] with an OSNet backbone [5] trained for re-identification on ImageNet [48], producing 512 dimensional full-body feature vectors and (iii) use camera calibrations (or ground-plane homographies based on availability) to project the center of the bottom edge of each bounding box, producing floor projections into a joint 2D coordinate system.

3.1 Problem Setting

Given a set $C = \{c_i\}_{i=1}^{n}$ of n cameras overlooking a scene with varying degrees of overlap, let each camera produce a frame $f_{c_i}^{t}$ at time t. Let $D_{c_i}^{t} = \{d_{c_i j}^{t}\}$ be the set of

object detections for frame $f_{c_i}^t$. Then, the goal of multi-camera multi-object tracking is to correctly assign all detections $d_{c_i j}^t$ in $D_{c_i}^t$ to some global track id $T_k = T_k^t$ where $T_k^t = \{d_{c_l j}^t\}_{l=1}^m$ is the set of detections belonging to the object with track id k at frame t.

In this work, we focus on the problem of person tracking which has the same problem formulation as the generalized formulation above with the addition that all detections are persons. Further, in our setting, we assume that the set of cameras C are overlooking a flat ground plane or a topology that can be approximated reasonably well with a flat plane. We also require that the set of cameras have a reasonable coverage of the scene, this is explained in detail in a following section.

Figure 3.1 provides an overview of our problem setting, given a calibrated multi-camera network with an arbitrary number of cameras, our goal is to produce globally consistent track IDs for all person detections at every frame.

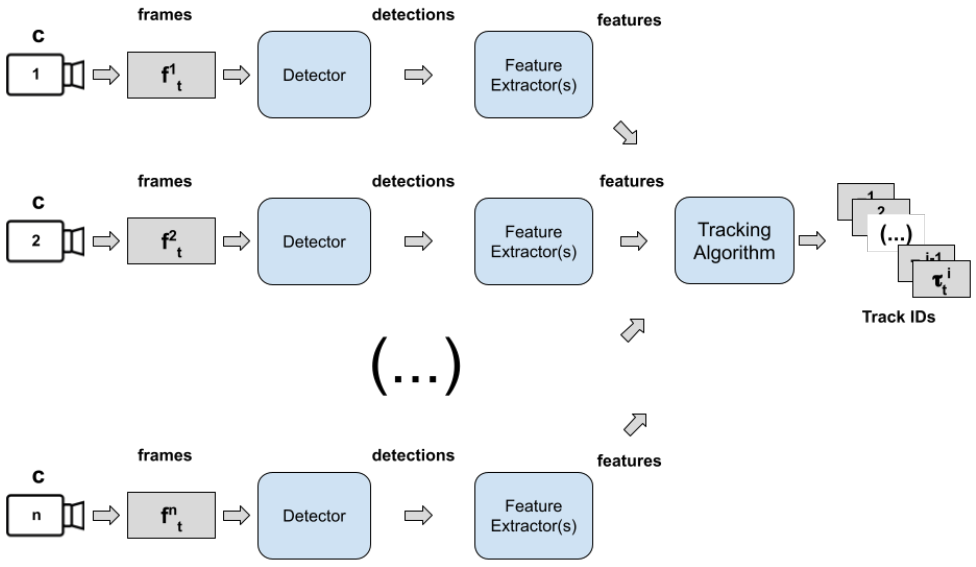

Figure 3.1: Overview of our approach. For an n camera setting, we begin with n parallel pipelines. Each pipeline extracts detections and corresponding features at each frame t. The joint tracking algorithm then computes globally consistent track IDs for each detection.

We explain each component of the Joint Tracker in detail in the following section.

3.2 Method

Given n synchronized cameras $c_i \in C$ streaming at the same FPS setting, let $f_j^{c_i}$ denote frame j for camera i. Given $f_j^{c_i}$, we pass it through some person detector D to produce a set $D_j^{c_i}$ of detections. Then, for every detection $d \in D_j^{c_i}$, we compute a set of features to be used in the decision making step. In our formulation, we use two types of features. Namely, floor-plane projections extracted using camera calibration information and visual feature vectors extracted using a neural network. Figure 3.2 serves as a high level diagram of the feature extraction module. Details of the feature extraction module are provided later in the chapter.

Given floor projection features $P_j^{c_i}$ and visual features $F_j^{c_i}$ for the set of detections $D_j^{c_i}$, we define our problem formulation around a cost matrix construction. At a high level, the main loop of our Joint Tracker is explained in Algorithm 1.

Algorithm 1: Joint Tracker Pseudocode

```
G ← {} // Instantiate empty gallery of global tracks
for each frame t do
    F ← {}                  // Instantiate empty feature set
    D ← {}                  // Instantiate empty detection set
    for each camera c do
        D_c ← Set of detections for camera c at time t
        F_c ← Features for D_c
        D.append(D_c)
        F.append(F_c)
    S ← group_detections(D, F)
    M ← calculate_cost_matrix(S, G)
    track_ids ← hungarian_matching(M)
    Update G with track_ids
```

The cost matrix is similar to ByteTrack [31] but extended to multiple cameras.

We now move on to explain the components of the main loop individually in detail beginning with the feature extraction module.

3.2.1 Features

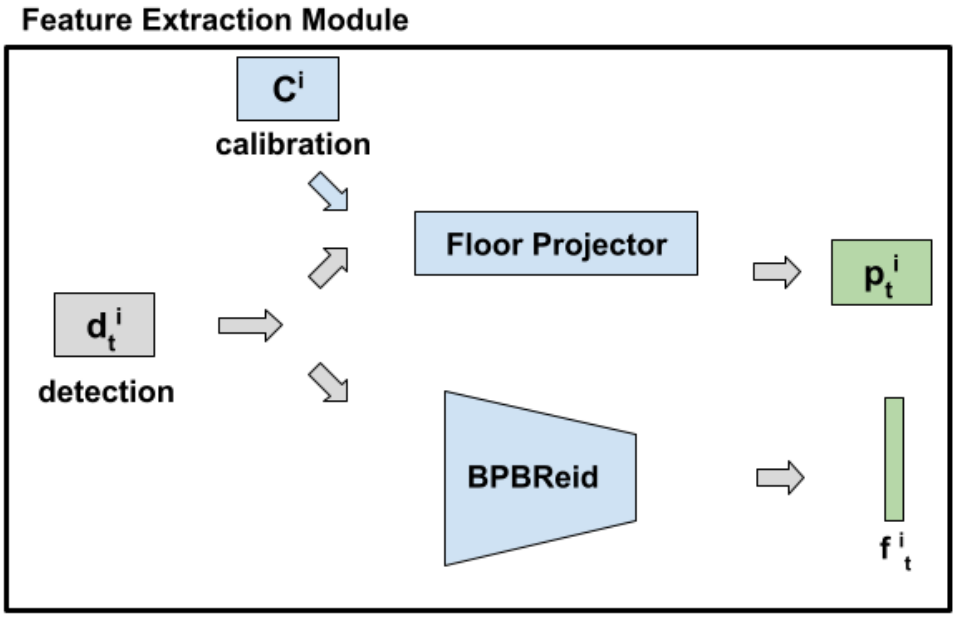

Figure 3.2: Overview of the feature extraction module in our formulation. Given an arbitrary detection d_i and calibrations C_i from the corresponding camera, the module extracts floor projection features and visual features using the respective extraction methods.

3.2.1.1 Visual Features

For visual feature extraction, we use a BPBReid (Body-part-based Re-Id) network [4] with an OSNet (Omni-Scale Network) [5] backbone pre-trained on open source person re-identification datasets Market1501 [49] and DukeMCMT [50]. The OSNet backbone is a convolutional neural network designed to extract features at various scales while BPBReid makes use of pose information at training time to extract body parts.

Given a variable sized person detection crop, we pad and resize is to $(384, 384)$ before feeding it to the feature extraction network. The network then produces a 3584 dimensional visual feature vector representing the crop.

3.2.1.2 Floor Projections

We assume camera calibrations are available in our setting. Camera calibrations are intrinsic and extrinsic matrices and optionally distortion parameters which define a mapping from any pixel on a given camera to a globally consistent floor coordinate system. If distortion parameters are ignored, an image to floor plane mapping can also be described by a homography matrix $H = IE$ where I and E are the intrinsic and extrinsic matrices of the target camera.

Given H_i or the pair (I_i, E_i) for some camera c_i, we can map any pixel on c_i to its corresponding global floor coordinate simply in the following way:

$$p_i = (c_i' \cdot I_i) \cdot E_i \tag{3.1}$$

or

$$p_i = c_i' \cdot H_i \tag{3.2}$$

For notational convenience, assume $c_i' = (x, y, 1)$ where (x, y) is the 2D image coordinate. Then $p_i = (p_i^x, p_i^y, 0.0)$ is the global floor coordinate corresponding to c_i. Note that both H_i and the pair I_i, E_i are invertible matrices as they define simple homographies between two planes.

Given a detection box d_j from camera c_i, let g_j be the center of the bottom edge of d_j. We call g_j the "floor contact point" of d_j. Then, the floor projection of d_j is computed as follows:

$$\text{floor\_projection}(d_j) = (g_j' \cdot I_i) \cdot E_i \tag{3.3}$$

The floor projection module therefore is an extremely lightweight matrix multiplication operation, mapping a given detection to a globally consistent coordinate we designate a floor projection point.

3.2.2 Detection Grouping

Given sets of detections D with corresponding floor projections, detection grouping is an algorithm used to create "person hypotheses" for the final matrix score calculation stage of the pipeline. Starting from some arbitrary camera, we produce detection groups based on floor projection distance. Two detections are assigned to the same group if a) they are from different cameras, b) they are within some radius r of each other. If multiple detections satisfy the constraint, we pick to lowest distance candidate.

We provide pseudocode for our implementation of the detection grouping algorithm in Algorithm 2.

Algorithm 2: Detection Grouping Pseudocode

Input: D `// Multi-camera detections at current frame.`
Input: λ `// Matching radius parameter.`
$P \leftarrow \{\}$ `// Instantiate empty gallery of detection groups`
foreach *detection* $d \in D$ **do**
  $P_d \leftarrow \{d\}$ `// Instantiate detection group for` d
  $d_p \leftarrow \{d' \in D \mid \text{dist}(d, d') < \lambda \text{ and } \text{cam}(d) \neq \text{cam}(d')\}$ `// Potential matches.`
  foreach $d' \in d_p$ **do**
    if no element in P_d is from the same camera as d' **then**
      P_d.append(d')
    else if there is an element d'' in P_d from the same camera as d' **then**
      $d_{app} \leftarrow \arg\min_{x \in \{d', d''\}} \text{dist}(d, x)$
      $P_d \leftarrow (P_d \setminus \{d''\}) \cup \{d_{app}\}$
  $P \leftarrow P \cup \{P_d\}$

The detection grouping stage produces person hypotheses consisting of sets of cross camera detections. We then move on to cost matrix calculation and feature fusion across detection groups.

3.2.3 Cost Matrix Calculation

Given a set of detection groups (person hypothesis) S and the current gallery of global track G, we construct a cost matrix to do Hungarian matching based assignment between the sets.

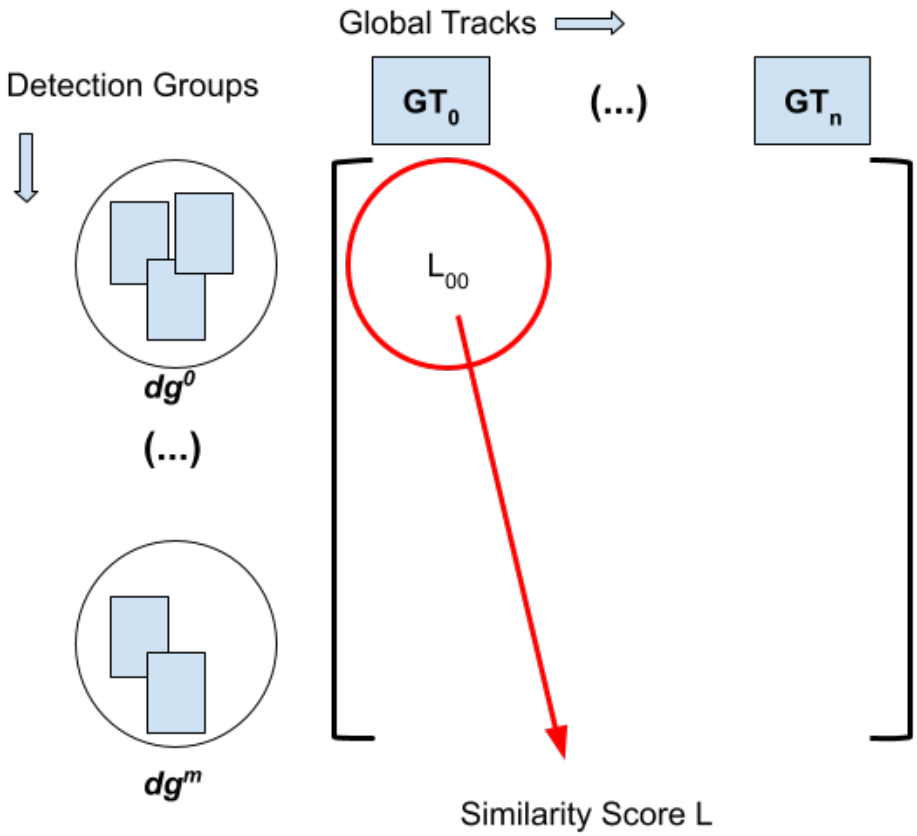

Figure 3.3: Cost matrix for the assignment problem. Rows correspond to detection groups and columns correspond to global tracks from the dictionary.

The total cost for a given entry of the matrix is as follows:

$$\lambda L_d(\text{mean}(GT_0), \text{mean}(dg^0)) + (1-\lambda)L_v(F(GT_0), F(dg^0)) \tag{3.4}$$

Where L_d represents the floor projection distance measurement, which is L2 distance on the floor plane converted to centimeters. We enforce L_d to be smaller than 1 meter for a matching and normalize it by 100, thus $\forall x_d(x) \in [0, 1]$. We calculate L_d between the mean floor projection point of the final frame of GT_0 and the mean floor

projection point of dg^0. L_v represents the visual distance, which is cosine distance in our implementation and thus normalized in $(0, 1)$. The function F is used to describe both the final frame of GT_0 and dg^0. Finally λ is a hyperparameter representing feature weight, with a higher λ value emphasizing floor projections.

We finally move on to describe a novel Multi-View Fusion (MVF) network for the function F. That is, for visual features, we propose a neural network to produce the representative feature instead of basic averaging across views.

3.2.4 Multi-View Fusion Network

We describe our Multi-View Fusion network proposal in this section. As previously mentioned, the MVF is proposed as an alternative to simple averaging when attempting to create summary vectors for detection groups. The network architecture is inspired by the Decoder module in Masked Autoencoders [51]. Similarly to the decoder module, the MVF fixed length masked sequences of feature vectors as input and consists of a series of transformer layers. We separate the training and inference procedures and explain them using figures 3.4 and 3.5.

At training time, we use a frozen pre-trained visual feature extractor (BPBReid in our experiments) to extract features vectors for all training crops. Then, we construct sets of possible positive examples using masking. Finally the masked sequences are fed through the MVF and the output descriptor feature vector is passed through a fully-connected layer. We train the network to predict the correct unique person ID with cross-entropy loss following the literature [4, 5]. Figure 3.4 provides an overview of this approach.

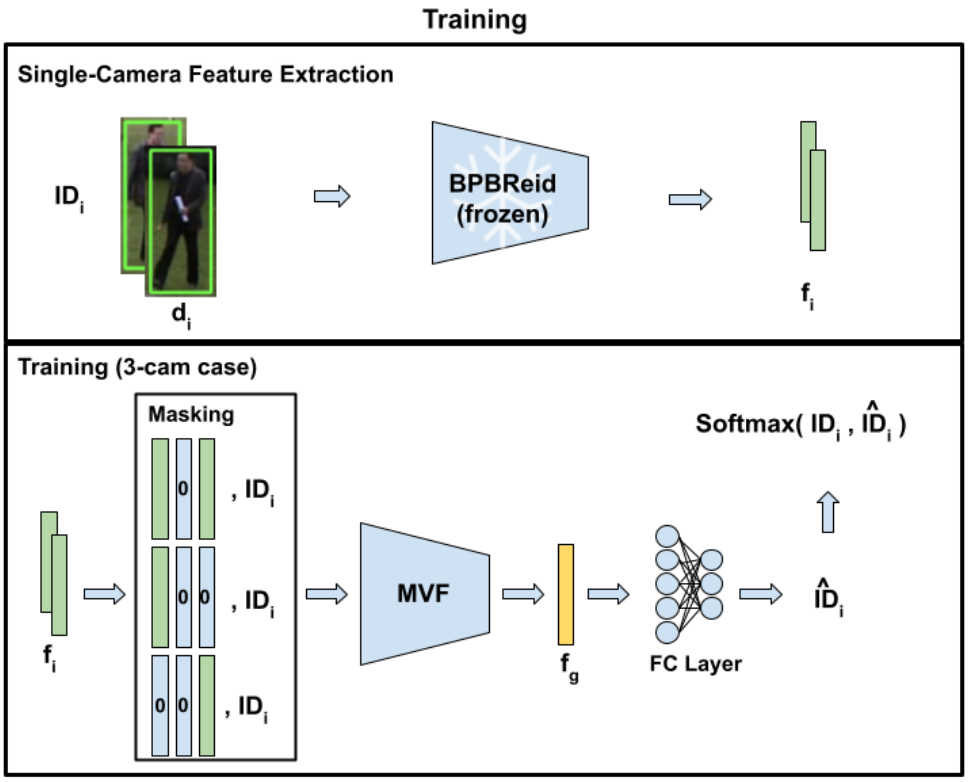

Figure 3.4: Training procedure for the Multi View Fusion network. A frozen pre-trained BPBReid [4] network with an OSNet [5] backbone is used to extract visual feature vectors which are used to generate positive training examples via masking. The MVF network is used to fuse feature vector sequences into a single descriptor vector which is then passed through a fully connected layer and softmax to arrive at a class ID. The network is trained with cross entropy loss using each unique person ID as a separate class following the literature.

After training, the final linear layer is removed and the MVF is used to produce the final descriptive feature vectors for a given detection group. The masking approach to generating training examples allows the network to generalize better to conditions with impaired visibility. We use fixed length sequences based on the number of cameras and sinusoidal positional encodings. Figure 3.5 provides an overview of this approach.

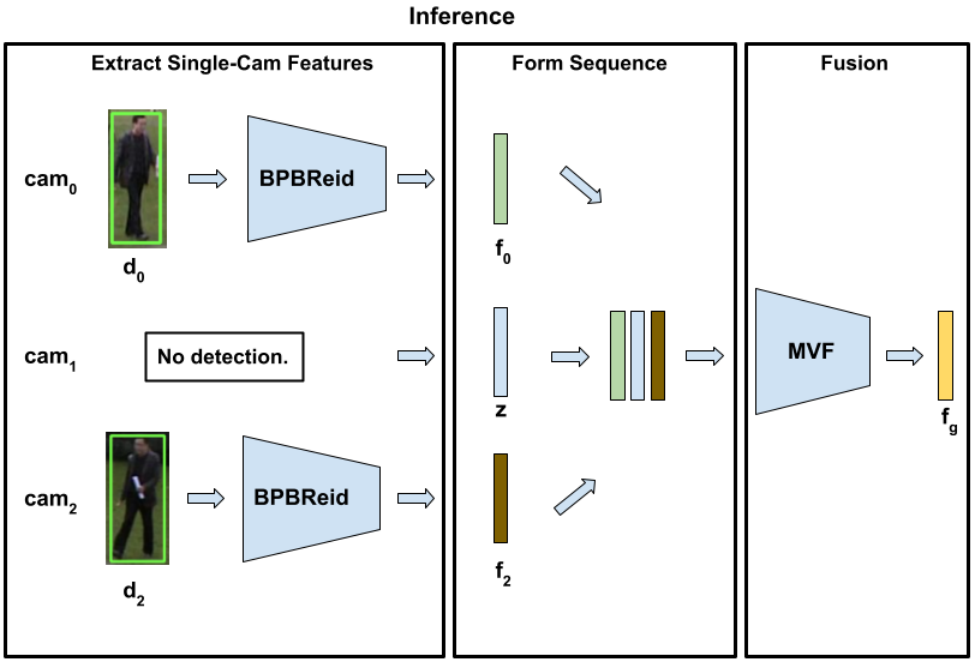

Figure 3.5: Inference procedure for the Multi View Fusion network. After training, the final fully-connected layer is dropped and the MVF output descriptor feature vector is used to represent the target detection group.

3.3 Assumptions and Implementation Details

We discuss our implementation details and assumptions in this chapter. Our main assumption is the existence of camera calibration parameters for our network of cameras. We assume both intrinsic and extrinsic parameters are available for each camera such that the parameters can be used to map any pixel coordinate on a camera to a 2D floor plane coordinate system shared by all cameras. We use floor projections for detection grouping and as global features for tracking as discussed earlier in the chapter. We use the OpenCV pinhole model with no distortion parameters in our experiments.

Next, we discuss camera coverage, a mostly overlooked but central issue for any tracking system. Most if not all studies make strong assumptions about camera positioning and coverage without explicitly addressing them. Earlier in the chapter, we briefly mention the term “reasonable coverage” to describe the ideal camera setting for our method. This section contains a more thorough explanation.

3.3.1 Camera Coverage

The term “reasonable coverage” is less straightforward to formally define than it might appear and has a full fledged field of research attached to it [52]. Studies in the field mostly focus on the optical camera placement (OCP) problem which can be formulated as a graph set-cover problem [53] where the goal is to optimize camera placement to minimize the number of cameras required for coverage. Expectably, the field has experimented with various definitions of “acceptable coverage” which we found to be fairly use-case specific in our studies.

In this study, we define acceptable camera coverage as the coverage of the entire scene of tracking by one or more cameras, with no blind zones. We further define “good coverage” as follows: Acceptable camera coverage conditions are satisfied, further, for all pairs of cameras with overlapping views, the overlapping regions on the image planes of the cameras are large enough to contain a full person detection. Figure 3.5 provides a visual representation of our coverage conditions and underlines the difference between acceptable and good coverage.

In our experiments, we observed that fragmentation errors predominantly occur at camera intersection regions during transitions. Interestingly, when these intersection areas are designed for reasonable visibility across multiple camera views, the occurrence of fragmentation errors significantly diminishes. This highlights the importance of strategic camera placement in minimizing fragmentation challenges in multi-camera tracking systems.

In our experiments, we find that fragmentation errors predominantly occur at camera intersections during transitions. Intersection regions with reasonable visibility in several camera views produce significantly fewer fragmentations. This observation is the motivation behind our “acceptable” versus “good” visibility standards. Luckily, most existing public multi-camera person tracking datasets satisfy both conditions in the vast majority of their scenes.

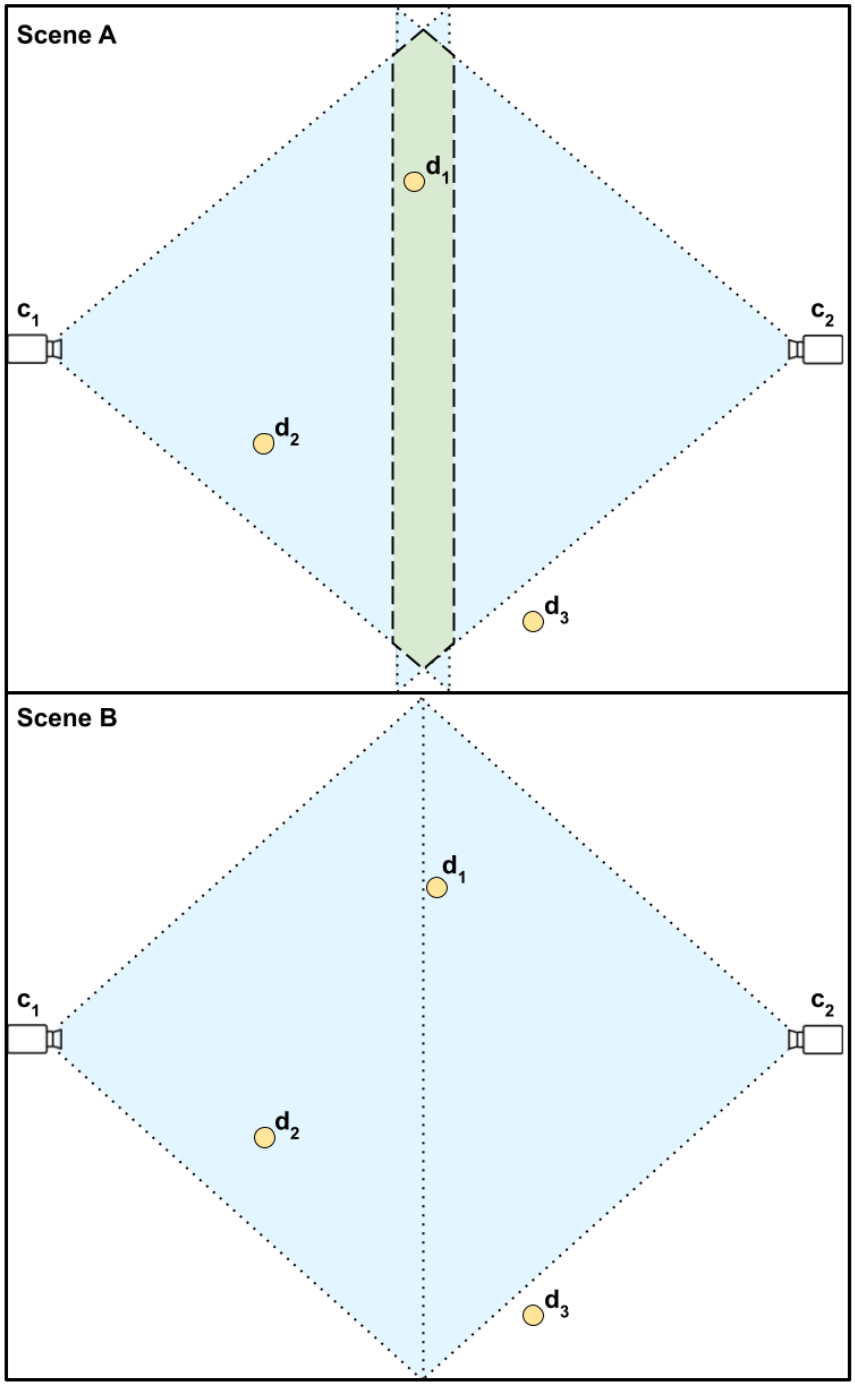

Figure 3.6: Birds-eye views of a two-camera system. Cyan represents visibility, green represents overlap and yellow points represent detections. Regions outlined by dotted lines satisfy our acceptable coverage criterion. Detection d_1 near the intersection is visible to both cameras on Scene A and only one camera at a time in Scene B, thus Scene A further satisfies the good visibility criterion.

CHAPTER 4

EXPERIMENTS

This chapter presents experimental results for the proposed methods on a variety of multi camera multi object tracking datasets from the pedestrian tracking domain. An overview of the datasets used in this study is followed by a thorough description of evaluation methods. Due to its complex definition, MCMOT has various evaluation metrics, focusing on different aspects of the problem. We provide explanations for the most commonly used evaluation methods in this section, which we also use.

4.1 Datasets

This section details the multi-camera multi-object tracking datasets used for experiments in this work. Since the study focuses on person tracking, all datasets are person tracking datasets. Due to our usage of person floor plane projections, we also require that the datasets have intrinsic and extrinsic camera calibrations.

WILDTRACK

The WILDTRACK [3] dataset is a multi-camera pedestrian tracking dataset consisting of 7 synchronized videos with each video having 400 annotated frames at 2 FPS. There are a total of 313 pedestrians in the dataset across various combinations of cameras. Following the literature [18, 3], we use the first 90% of the annotated frames for training and evaluate our results on the last 10%. This corresponds to around 140 global tracks, 600 local tracks and 5500 detection boxes across all cameras. WILDTRACK also contains intrinsic and extrinsic camera parameters which can be used

for computing person floor projections.

EPFL Multi Camera Pedestrians Dataset

The EPFL Multi Camera Pedestrians Dataset [3] is a collection of multi-camera pedestrian tracking datasets containing annotated videos and camera calibrations. We focus on the Terrace sequence which consists of around 3.5 minutes of synchronized video across 4 cameras collected at 25 FPS (5010 frames per camera). There are a total of 7 pedestrians in the videos, though the number of global tracks is closer to 40 due to targets leaving and re-entering the scene, this translates to 100 local tracks and 45000 detections per camera. Following [18, 54], we use the first 10% of the frames for training and the remaining frames for testing.

PETS09 Dataset

The PETS09 dataset is a collection of multi-camera pedestrian tracking datasets collected from 5 synchronized cameras at 7 FPS. We focus on the S2-L1 sequence which consists of 795 annotated frames containing 19 global tracks with around 20 local tracks and 8000 detections per camera. We use the first 50% of the dataset for training and test on the remaining 50% following [18, 55].

4.2 Evaluation Methods

This section details the quantitative evaluation methods and metrics used in this study.

MOTA (Multi-Object Tracking Accuracy):

$$\text{MOTA} = 1 - \frac{\sum_t(\text{FN}_t + \text{FP}_t + \text{IDS}_t)}{\sum_t(\text{GT}_t)} \tag{4.1}$$

Where FN represent the false negatives, FP false positives, IDS ID-switches and GT ground truth detections. MOTA [56] measures the percentage of mistakes made by the tracker, represented as a sum of missed, falsely identified or switched detections. Note that each ID switch is penalized once per camera in the multi-camera setting.

MOTP (Multi-Object Tracking Precision):

$$\text{MOTP} = 1 - \frac{\sum_t (s_t)}{\sum_t (|s_t|)} \tag{4.2}$$

where s_t represents the summed detection to ground truth distance between each matching (this can be an integer or floating point number depending on distance calculation) at time t and $|s_t|$ represents the total number of ground truth matchings made at time t. Note that MOTP measures localization on the image coordinate system and is therefore a mainly single-camera object detector localization metric.

We now define the Mostly Tracked and Mostly Lost metrics from [57].

MT (Mostly Tracked):

A track is considered "Mostly Tracked" if 80% or more of its frames have been correctly identified. This metric is the percentage of tracks in the dataset that are "Mostly Tracked".

ML (Mostly Lost):

A track is considered "Mostly Lost" if 20% or fewer of its frames have been correctly identified. This metric is the percentage of tracks in the dataset that are "Mostly Lost".

We now define IDF1, or Identification F1 proposed in [58]. We first make a series of definitions.

ID-Recall (IDR):

$$\text{ID-Recall (IDR)} = \frac{\text{IDTP}}{\text{IDTP} + \text{IDFN}} \tag{4.3}$$

Where IDTP is the number of true identifications and IDFN is the number of false identifications made by the tracker. IDR measures the recall of the tracker, that is, what percentage of the total number of ground truth detections it identifies correctly.

ID-Precision (IDP):

$$\text{ID-Precision (IDP)} = \frac{\text{TP}}{\text{TP} + \text{FP}} \tag{4.4}$$

Where TP is the number of true identifications and FN is the number of false identifications made by the tracker. IDP measures the precision of the tracker, that is, the number of correct identifications over all identifications made.

Identification F1 (IDF1):

$$\text{Identification F1} = \frac{\text{TP}}{\text{TP} + \frac{1}{2}(\text{FN} + \text{FP})} \tag{4.5}$$

IDF1 [59] attempts to measure both the precision and recall of the tracker. When extended to the multi-camera setting, IDF1 treats all detections equally regardless of track continuity. That is, IDF1 only indirectly measures ability to continuously track a target across its entire trajectory.

We find that track continuity is crucial for many applications of global tracking from security [60, 22, 23], to self-driving vehicles [6] and many more. In our formulation, a track is considered continuous if its target is correctly identified on at least one camera throughout its journey in the tracking zone. In the following section, we propose a new global tracking performance metric to better capture global tracking continuity.

4.2.1 Global Identification Metrics

We propose a new series of evaluation metrics we call Global Identification Metrics to better capture global track continuity in contrast to IDF1, which operates on all cameras separately and equally with no regard for global continuity. Our main metric is Global-IDF1 (GIDF1), designed to better capture a given algorithms ability to satisfy the following goal; the correct and continuous tracking and re-identification of an objects entire trajectory within the tracking scene. Our proposed metrics are generalizations of IDF1 and accompanying metrics to the multi-camera setting.

We propose that while correctly re-identifying all detections is important, continuously tracking a target's entire journey is often just as important as it serves to provide downstream tasks with trajectories in many applications. We extend IDF1 to Global IDF1 (GIDF1) in this section.

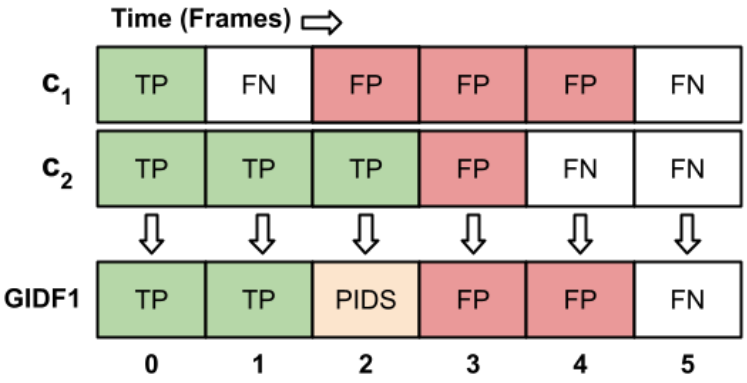

Figure 4.1: All possible tracker correctness cases for a 2-camera system. We construct the definitions for a general n-camera setting without loss of generality.

4.2.1.1 Basic Definitions

Based on Figure 4.1, we make the following series of definitions:

True Positive (TP):

The target is correctly identified on at least one camera and not incorrectly identified on any camera. Cases 0 and 1 from Figure 4.1 are examples of TPs for a 2-camera setting.

Partial ID-Switch (PIDS)

The target is correctly identified on at least one camera and incorrectly identified on at least one camera. Case 2 from Figure 4.1 is an example of a PIDS.

False Positive (FP):

The target is incorrectly identified on at least one camera and not correctly identified on any cameras. Cases 3 and 4 from Figure 4.1 are examples of FPs for a 2-camera setting.

False Negative (FN):

The target is missed on all cameras. Case 5 from figure 4.1 is an example of a FN.

4.2.1.2 Definitions

Based on the basic definitions from the previous subsection, we now define the Global Identification metrics.

Global ID-Recall:

$$\text{Global ID-Recall (GIDR)} = \frac{\text{TP} + \text{PIDS}}{\text{TP} + \text{FN} + \text{PIDS}} \tag{4.6}$$

GIDR measures the ability of a global tracker to correctly identify a tracking target on at least one camera at any given frame.

Global ID-Precision (GIDP):

$$\text{Global -ID Precision} = \frac{\text{TP}}{\text{TP} + \text{FP} + \text{PIDS}} \tag{4.7}$$

GIDP measures the correctness of a global tracker's predictions. That is, the ratio of fully correct detection groups produced by the tracker to all groups.

Global IDF1 (GIDF1):

$$\text{GIDF1} = \frac{\text{TP} + \frac{1}{2}\text{PIDS}}{\text{TP} + \text{PIDS} + \frac{1}{2}(\text{FP} + \text{FN})} \tag{4.8}$$

Similarly to the original per-detection IDF1 [58], this measures a combination of precision and recall for the global formulation.

4.3 Quantitative Results

This section contains quantitative results for the proposed algorithm and comparisons to various benchmarks. The results are produced using the evaluation methods on the datasets detailed in the previous sections of this chapter.

Here we compare our model to baseline solutions on EPFL-Terrace, PETS09-S2L1 and WILDTRACK. KSP is a simple [61] K-Shortest Path approach (graph based).

HCT [34] uses a Hierarchical Composition of Tracklets and TRACTA [62] uses a Restricted Non-negative Matrix Factorization approach for tracklet matching. LMPG (or Lifted-Multicut) uses several specialized neural networks with a graph formulation to fix and connect tracklets. Note that all of these methods are based on tracklet matching are thus window based or offline.

Table 4.1: Comparisons to baseline methods on the EPFL [1] dataset, Terrace1 sequence to our method. Results taken as reported from the original papers.

Method	**MOTA**	**MOTP**	**MT**	**ML**
KSP [61]	58	63	-	-
HCT [34]	72	71	-	-
TRACTA [62]	**81**	79.5	-	-
Ours + Simple Averaging	78.09	**87.32**	100.0	0.0
Ours + MVF	71.8	**83.6**	89.6	0.0

Table 4.2: Comparisons to baseline methods on the PETS09 dataset, S2-L1 sequence [2] to our method. Results taken as reported from the original papers.

Method	**MOTA**	**MOTP**	**MT**	**MP**
KSP [61]	80	57	-	-
HCT [34]	89	73	-	-
TRACTA [62]	87.5	79.2	-	-
LMGP [18]	**97.8**	82.4	**100.0**	**0.0**
Ours + Simple Averaging	81.38	**86.43**	94.73	**0.0**
Ours + MVF	83.4	**86.0**	**100.0**	**0.0**

We note that using the Multi-View Fusion network results in a significant decrease in performance for the EPFL Terrace sequence, while there is a small increase in performance for the PETS09 dataset. We attribute this to training data availability as the network fails to train to convergence for very small datasets. Our experiments show a positive correlation between dataset size (total number of global tracks/cameras) and performance increase introduced by the MVF network. We further discuss this in the following section.

Some of our baseline models for EPFL and PETS09 do not report results for WILDTRACK. For completeness, we use several other available tracking solutions as baseline for WILDTRACK. Namely, we use KSP-DO [3] is an modification of KSP which uses Deep Occlusion Reasoning [63] to estimate person floor projections and occlusions. GLMB [33] uses a 3D occlusion model in a Bayesian formulation to handle occlusions, misdetections, clutters etc. GLMB is an online solution capable of producing track ids in real-time. GLMB can also be paired with Deep Occlusion reasoning to produce improved results. DMCT (Deep Multi-Camera Tracker) [32] is also a real-time solution making use of a series of neural networks for 3D occupancy estimation and tracking.

Table 4.3: Comparisons of baseline tracking solutions on the WILDTRACK [3] dataset to our method. KSP and TRACTA results were obtained by using their respective codebases as WILDTRACK results were not published. Remaining results were taken as reported from the original papers.

Method	**IDF1**	**IDP**	**IDR**	**MOTA**	**MT**	**ML**	**GIDF1**
KSP [61]	51.7	64.3	41.3	48.3	5.1	43.2	55.2
KSP-DO [3]	73.2	83.8	65	69.6	28.7	25.1	-
GLMB-DO [33]	72.5	82.7	72.2	70.1	-	-	-
DMCT [32]	81.9	81.6	82.2	74.6	65.9	4.9	-
TRACTA [62]	88.1	88.9	87.5	82.2	77.4	1.6	88.4
LMGP [18]	**98.2**	**99.3**	**97.2**	**97.1**	**97.6**	**1.3**	-
Ours + Simple Averaging	85.2	90.1	80.7	78.1	80.2	2.2	90.2
Ours + MVF	86.8	91.6	82.5	78.9	81.8	2.0	**90.5**

Given the more reasonable amount of training data (360 frames, 7 cameras) for the WILDTRACK dataset, we observe an improvement introduced by the MVF network. This expected behavior is hypothesized scale up to larger datasets.

4.3.1 Evaluation on Global ID Metrics

This section contains experimental results evaluated using Global ID metrics for the WILDTRACK dataset. Official codebases were used to evaluate the baseline models.

Table 4.4: Comparisons of baselines to our method using the Global ID metrics on the WILDTRACK [3] dataset.

Method	**IDF1**	**IDP**	**IDR**	**GIDF1**	**GIDP**	**GIDR**
KSP [61]	51.7	64.3	41.3	55.2	57.0	53.3
TRACTA [62]	**88.1**	88.9	**87.5**	88.4	**85.7**	90.9
Ours + Simple Averaging	85.2	90.1	80.7	90.2	85.1	95.4
Ours + MVF	86.8	**91.6**	82.5	**90.5**	85.6	**96.1**

We note that GIDR is generally higher than its single camera counterpart. This is due to the more relaxed definition designed to better capture the multi-camera nature of the tracking problem in our setting. For use cases favoring the correct tracking of a target's trajectory across a multi-camera network, GIDR is more informative than its single-camera counterpart which requires that the target is correctly identified on all cameras at every frame.

4.4 Ablation Studies

This section contains ablation studies aimed to better explore the strengths and weaknesses of the various components of our solution. We provide experimental results with different settings. We carry out two series of experiments, first, we remove the geometric and visual feature components of our cost matrix function and compare the results to the original joint formulation to better analyze the effects of each component. For the second series of experiments, we remove the multi-view fusion network from our system, replacing it with simple averaging, and compare the results. We conduct all combinations of both series of experiments for completeness. Experimental results are available in Table 4.5.

In our first series of experiments, to compare the effectiveness of our features, we

conduct experiments on the WILDTRACK dataset by removing the geometric and visual components at the cost matrix calculation stage and comparing the results to the joint cost formulation. Table 4.5 contains the results of our quantitative analysis. It can be seen from the table that removing either component results in a dramatic decrease in tracking performance across all metrics. We can also see that our algorithm performs even more poorly with only the visual component, demonstrating the power of floor projections as a geometric feature in multi-camera tracking.

Table 4.5: Evaluations of our method with various ablations on the WILDTRACK [3] dataset. The keyword "mean" represents simple averaging of feature vectors while MVF means the Multi-View Fusion network was used to generate descriptive features for detection groups.

Method	**IDF1**	**IDP**	**IDR**	**MOTA**	**MT**	**ML**	**GIDF1**	**GIDP**	**GIDR**
Geometric	71.8	75.3	67.8	63.3	66.5	7.3	78.3	74.3	84.1
Visual (mean)	59.1	58.4	60.7	54.1	42.2	12.3	60.6	57.5	63.4
Visual (MVF)	63.5	59.2	62.0	55.7	47.2	10.9	65.1	59.2	71.3
Joint (mean)	85.2	90.1	80.7	78.1	80.2	2.2	90.2	85.1	95.4
Joint (MVF)	**86.8**	**91.6**	**82.5**	**78.9**	**81.8**	**2.0**	**90.5**	**85.6**	**96.1**

In the second series of experiments, we remove our multi-view fusion network in favor of simple averaging and compare the results. Rows 2 through 5 of Table 4.5 show that the MVF introduces improvements across the board, being especially useful in the visual-feature-only setting. We attribute this to the multi-camera representation power of the MVF compared to simple averaging. We hypothesize that geometric features account for most of the multi-camera representation power of our system in the joint setting. Without geometric features, we see a dramatic increase in performance introduced by the MVF filling in the multi-camera representation gap.

The formulations from our experiments in Table 4.5 have strong implications for run time as computationally heavy components are removed in some settings. We report resulting changes in run times in the following section.

4.5 Computational Complexity and Run Time Analysis

This section contains a computational complexity analysis of the proposed algorithm. We also provide tables containing average run time measurements of the algorithm both component-wise and end-to-end to provide a better understanding of the effects of various settings on inference time. We demonstrate that the algorithm is capable of producing track IDs in real time for a 7-camera network covering a crowded scene with 20 visible people on average per camera per frame and more than 100 unique individuals using the WILDTRACK [3] dataset. We also show that a floor projection only version of our algorithm can achieve much faster run time, with implications for use on much larger camera networks.

4.5.1 Computational Complexity

The proposed algorithm has a computational complexity of $O(\max(n, m)^3)$ where n is the number of detections at frame t and m is the number of global tracks in the dictionary. This is due to the linear sum assignment function call. In contrast to the exponential time worst case formulations used in most contemporary graph based offline or windowed algorithms, we offer a polynomial time worst case solution.

In practice, due to n and m being small (<100) numbers, the bottleneck becomes the visual feature extraction network. The best performance is achieved when only floor projections are used for tracking.

4.5.2 Run Time Analysis

In this section we provide an analysis of runtime for the algorithm. The experiments are made using a machine with 8 (Intel Xeon) vCPU cores and a Nvidia Tesla V100 GPU. We provide run times for each component of the algorithm running on the 7-camera WILDTRACK dataset with an average of 20 people per camera on each frame, the results are averaged over 400 frames of inference.

Table 4.6: Component level breakdown of run times for the full tracking pipeline. We use the WILDTRACK dataset for this analysis where there are up to 20 people per camera for 7 cameras. VFE stands for Visual Feature Extraction, VF for Visual Features, FP for Floor Projections and MVFNet is our novel multi-view fusion network.

Component	**Run Time (1 time-step)**	**FPS**
VFE w BPBNet, OSNet backbone	294.6 ms	3.4
VFE w BPBNet, OSNet backbone (Parallelized)	51.2 ms	19.5
Floor projections	1.1 ms	909
Floor projections (Parallelized)	0.7 ms	1428
Group detections (Averaging)	1.8 ms	556
Group detections (MVFNet)	21.3 ms	47.0
Group detections (FP only)	1.7 ms	588
Calculate cost matrix	0.1 ms	10000
Hungarian matching & Update	3.2 ms	312.5

It can be seen from Table 4.6 that the bottleneck for our algorithm is by far the visual feature extractor. Large run time improvements can be achieved by replacing the visual feature extractor with a faster model or by disabling it completely. The second longest run time component is the multi-view fusion network MVFNet, which introduces up to 20 ms of extra run time per time step in our pipeline. This component could be turned off in favor of averaging or due to visual features being turned off. We observe that the algorithm achieves fastest run times when visual features are turned off and the slowest run times with visual features and MVFNet enabled.

We can also clearly see from Table 4.6 that feature extraction stage run time dramatically improves with parallelization especially for visual feature extraction. Since our algorithm does not require cross camera information at the feature extraction stage, we can fully parallelize it for best run times without loss of performance. We further analyze end to end run times in Table 4.7.

It can be seen from Table 4.7 that when parallelized, the model achieves real-time status. Our best performing pipeline in terms of IDF1 and GIDF1 uses both visual features and floor projections and uses the MVFNet for fusion, we report 13 FPS

Table 4.7: End to end run times for the full tracking pipeline for various component combinations. We use the WILDTRACK dataset for this analysis where there are up to 20 people per camera for 7 cameras. VF stands for Visual Features, FP for Floor Projections and MVFNet is our novel multi-view fusion network. We report the best tracking performance in terms of IDF1 and GIDF1 when using both VF and FP with MVFNet.

Setting	**Run Time (1 time-step)**	**FPS**
VF and FP (Sequential), Averaging	300.8 ms	3.32
VF and FP (Parallel), Averaging	57.0 ms	17.54
VF and FP (Parallel), MVFNet	76.5 ms	13.07
FP only (Parallel)	5.7 ms	175.43

inference time at these settings. The table also shows that disabling visual features and running only on floor projections result in a huge decrease in run time, achieving up to 175 FPS on a crowded 7-camera dataset (WILDTRACK).

CHAPTER 5

CONCLUSIONS

In this study, we present a joint multi-camera multi-person tracker capable of producing track IDs at each frame in real time without dependency on a time-window of detections. We show that person ground contact points can be used as robust multi-camera features in a calibrated camera network. We conduct a detailed analysis of camera coverage as an invisible assumption in tracking problem statements and clearly describe our coverage requirements. We propose a novel multi-camera visual feature fusion network and show its effectiveness over naive averaging based fusion for larger datasets with more cameras and diverse viewpoints. We quantitatively evaluate our work on EPFL, PETS09 and WILDTRACK datasets and compare it to available baselines and state of the art. We address shortcomings in widely adopted tracking metrics by proposing a new metric designed to emphasize track continuity in a multi-camera network instead of weighing all ID assignments equally for all cameras at all frames. We evaluate and compare our method's performance using our newly developed metric. We finally provide run time and performance comparisons for our method using different combinations of features and feature fusion methods.

5.1 Limitations and Future Work

Our method depends on intrinsic and extrinsic camera calibrations used to map person detections to a common floor plane. We use the floor plane coordinates for detection grouping across cameras and in the final track ID assignment stage. This creates a limitation for scenes where camera calibration may be difficult. A potential direction for future improvement can be the addition of an automatic camera calibration step

using highly confident feature matchings or even person re-identifications similarly to Lee et al. [64] who propose a pose estimation based extrinsic calibration method. The integration of this capability to the tracking pipeline would relax the calibration dependency to a much easier to satisfy planar ground surface dependency.

Another direction for future research can be the improvement of the multi-view fusion network. The proposed network is trained for each scene/dataset separately since we use a fixed number of cameras and positional encodings during training. This introduces two main complications, namely, limited availability of training data for multi-camera tracking and the requirement for training on a scene before inference in contrast to the visual feature extractor which we can simply freeze and use in a different scene. If a scene independent version of the network can be developed, this would both significantly increase the amount of available training data and remove the requirement for training before inference on a new scene.

REFERENCES

[1] 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, 2008.

[2] J. Ferryman and A. Shahrokni, "Pets2009: Dataset and challenge," in *2009 Twelfth IEEE International Workshop on Performance Evaluation of Tracking and Surveillance*, pp. 1–6, 2009.

[3] T. Chavdarova, P. Baqué, S. Bouquet, A. Maksai, C. Jose, T. Bagautdinov, L. Lettry, P. Fua, L. Van Gool, and F. Fleuret, "Wildtrack: A multi-camera hd dataset for dense unscripted pedestrian detection," in *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 5030–5039, 2018.

[4] V. Somers, C. D. Vleeschouwer, and A. Alahi, "Body part-based representation learning for occluded person re-identification," 2022.

[5] K. Zhou, Y. Yang, A. Cavallaro, and T. Xiang, "Omni-scale feature learning for person re-identification," *CoRR*, vol. abs/1905.00953, 2019.

[6] D. Gragnaniello, A. Greco, A. Saggese, M. Vento, and A. Vicinanza, "Benchmarking 2d multi-object detection and tracking algorithms in autonomous vehicle driving scenarios," *Sensors*, vol. 23, no. 8, 2023.

[7] L. Fei and B. Han, "Multi-object multi-camera tracking based on deep learning for intelligent transportation: A review," *Sensors*, vol. 23, no. 8, 2023.

[8] A. Rangesh and M. M. Trivedi, "No blind spots: Full-surround multi-object tracking for autonomous vehicles using cameras and lidars," *IEEE Transactions on Intelligent Vehicles*, vol. 4, no. 4, pp. 588–599, 2019.

[9] J. Park, J. Hong, W. Shim, and D.-J. Jung, "Multi-object tracking on swir images for city surveillance in an edge-computing environment," *Sensors*, vol. 23, p. 6373, July 2023.

[10] Y. Jang, I. Jeong, M. Younesi Heravi, S. Sarkar, H. Shin, and Y. Ahn, "Multi-camera-based human activity recognition for humanndash;robot collaboration in construction," *Sensors*, vol. 23, no. 15, 2023.

[11] A. Sharath Chandra, M. Plasch, C. Eitzinger, and B. Rinner, "Context enhanced multi object tracker for human robot collaboration," in *Proceedings of the Companion of the 2017 ACM/IEEE International Conference on Human-Robot Interaction*, HRI '17, ACM, Mar. 2017.

[12] F. Ferraguti, C. Talignani Landi, S. Costi, M. Bonfè, S. Farsoni, C. Secchi, and C. Fantuzzi, "Safety barrier functions and multi-camera tracking for human–robot shared environment," *Robotics and Autonomous Systems*, vol. 124, p. 103388, Feb. 2020.

[13] Y. Zhang and T. E. Doyle, "Integrating intention-based systems in human-robot interaction: a scoping review of sensors, algorithms, and trust," *Frontiers in Robotics and AI*, vol. 10, 2023.

[14] B. Yang and R. Nevatia, "An online learned crf model for multi-target tracking," in *2012 IEEE Conference on Computer Vision and Pattern Recognition*, pp. 2034–2041, 2012.

[15] C. Dicle, O. I. Camps, and M. Sznaier, "The way they move: Tracking multiple targets with similar appearance," in *2013 IEEE International Conference on Computer Vision*, pp. 2304–2311, 2013.

[16] A. Dehghan, S. Modiri Assari, and M. Shah, "Gmmcp tracker: Globally optimal generalized maximum multi clique problem for multiple object tracking," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2015.

[17] W. Chen, L. Cao, X. Chen, and K. Huang, "An equalized global graph model-based approach for multicamera object tracking," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 27, no. 11, pp. 2367–2381, 2017.

[18] D. M. H. Nguyen, R. Henschel, B. Rosenhahn, D. Sonntag, and P. Swoboda, "LMGP: lifted multicut meets geometry projections for multi-camera multi-object tracking," *CoRR*, vol. abs/2111.11892, 2021.

[19] H. W. Kuhn, "The hungarian method for the assignment problem," *Naval Research Logistics Quarterly*, vol. 2, p. 83–97, Mar. 1955.

[20] J. W. Lee, M. S. Kim, and I. S. Kweon, "A kalman filter based visual tracking algorithm for an object moving in 3d," in *Proceedings 1995 IEEE/RSJ International Conference on Intelligent Robots and Systems. Human Robot Interaction and Cooperative Robots*, vol. 1, pp. 342–347 vol.1, 1995.

[21] Y. Du, J. Wan, Y. Zhao, B. Zhang, Z. Tong, and J. Dong, "Giaotracker: A comprehensive framework for mcmot with global information and optimizing strategies in visdrone 2021," 2022.

[22] B. Gaikwad and A. Karmakar, "Smart surveillance system for real-time multi-person multi-camera tracking at the edge," *Journal of Real-Time Image Processing*, vol. 18, 12 2021.

[23] A. Sharma, S. Anand, and S. K. Kaul, "Intelligent querying for target tracking in camera networks using deep q-learning with n-step bootstrapping," *Image and Vision Computing*, vol. 103, p. 104022, 2020.

[24] Y. Yoon, A. Kosaka, and A. C. Kak, "A new kalman-filter-based framework for fast and accurate visual tracking of rigid objects," *IEEE Transactions on Robotics*, vol. 24, no. 5, pp. 1238–1251, 2008.

[25] E. Cuevas, D. Zaldivar, and R. Rojas, "Kalman filter for vision tracking," *Measurement*, vol. 33, 01 2005.

[26] N. O. Salscheider, "Object tracking by detection with visual and motion cues," *CoRR*, vol. abs/2101.07549, 2021.

[27] W. Chen, X. Guo, X. Liu, E. Zhu, and J. Yin, "Appearance changes detection during tracking," in *2016 23rd International Conference on Pattern Recognition (ICPR)*, pp. 1821–1826, 2016.

[28] S. Yang, Y. Xie, P. Li, H. Wen, H. Luo, and Z. He, "Visual object tracking robust to illumination variation based on hyperline clustering," *Information*, vol. 10, no. 1, 2019.

[29] L. Leal-Taixé, "Multiple object tracking with context awareness," 11 2014.

[30] A. Dehghan, S. Modiri Assari, and M. Shah, "Gmmcp tracker: Globally optimal generalized maximum multi clique problem for multiple object tracking," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 4091–4099, 2015.

[31] 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," 2022.

[32] Q. You and H. Jiang, "Real-time 3d deep multi-camera tracking," *CoRR*, vol. abs/2003.11753, 2020.

[33] J. Ong, B. Vo, B. Vo, D. Y. Kim, and S. Nordholm, "A bayesian 3d multi-view multi-object tracking filter," *CoRR*, vol. abs/2001.04118, 2020.

[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 (CVPR)*, pp. 4256–4265, Jun 2016.

[35] W. Hu, X. Li, W. Luo, X. Zhang, S. Maybank, and Z. Zhang, "Single and multiple object tracking using log-euclidean riemannian subspace and block-division appearance model," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 34, no. 12, pp. 2420–2435, 2012.

[36] L. Zhang and L. van der Maaten, "Structure preserving object tracking," in *2013 IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1838–1845, IEEE, 2013.

[37] L. Zhang and L. van der Maaten, "Preserving structure in model-free tracking," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 36, no. 4, pp. 756–769, 2014.

[38] J. Yang, M. Gao, Z. Li, S. Gao, F. Wang, and F. Zheng, "Track anything: Segment anything meets videos," 2023.

[39] A. Kirillov, E. Mintun, N. Ravi, H. Mao, C. Rolland, L. Gustafson, T. Xiao, S. Whitehead, A. C. Berg, W.-Y. Lo, P. Dollár, and R. Girshick, "Segment anything," *arXiv:2304.02643*, 2023.

[40] J. Xing, H. Ai, and S. Lao, "Multi-object tracking through occlusions by local tracklets filtering and global tracklets association with detection responses," in *2009 IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1200–1207, IEEE, 2009.

[41] W. Luo, X. Zhao, and T. Kim, "Multiple object tracking: A review," *CoRR*, vol. abs/1409.7618, 2014.

[42] K.-S. Yang, Y.-K. Chen, T.-S. Chen, C.-T. Liu, and S.-Y. Chien, "Tracklet-refined multi-camera tracking based on balanced cross-domain re-identification for vehicles," in *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, pp. 3978–3987, 2021.

[43] Q. You and H. Jiang, "Real-time 3d deep multi-camera tracking," *CoRR*, vol. abs/2003.11753, 2020.

[44] D. Mendes, S. Correia, P. Jorge, T. Brandão, P. Arriaga, and L. Nunes, "Multi-camera person re-identification based on trajectory data," *Applied Sciences*, vol. 13, no. 20, 2023.

[45] B. Scassellati, "Towards a theory of intention recognition for human-robot interaction," *Autonomous Robots*, vol. 15, no. 1, pp. 1–18, 2003.

[46] R. A. Brooks, *Robots with Internal Models: The Dynamics of Intentionality*. MIT Press, 1991.

[47] S. Yang, F. Ding, P. Li, and S. Hu, "Distributed multi-camera multi-target association for real-time tracking," *Sci. Rep.*, vol. 12, p. 11052, June 2022.

[48] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "Imagenet: A large-scale hierarchical image database," in *2009 IEEE conference on computer vision and pattern recognition*, pp. 248–255, Ieee, 2009.

[49] L. Zheng, L. Shen, L. Tian, S. Wang, J. Wang, and Q. Tian, "Scalable person re-identification: A benchmark," in *Computer Vision, IEEE International Conference on*, 2015.

[50] E. Ristani, F. Solera, R. S. Zou, R. Cucchiara, and C. Tomasi, "Performance measures and a data set for multi-target, multi-camera tracking," *CoRR*, vol. abs/1609.01775, 2016.

[51] K. He, X. Chen, S. Xie, Y. Li, P. Dollár, and R. B. Girshick, "Masked autoencoders are scalable vision learners," *CoRR*, vol. abs/2111.06377, 2021.

[52] M. Song, D. Tao, and S. Maybank, "Sparse camera network for visual surveillance – a comprehensive survey," 02 2013.

[53] J. Kritter, M. Brévilliers, J. Lepagnot, and L. Idoumghar, "On the optimal placement of cameras for surveillance and the underlying set cover problem," *Applied Soft Computing*, vol. 74, pp. 133–153, 2019.

[54] S. Tang, M. Andriluka, A. Milan, K. Schindler, S. Roth, and B. Schiele, "Learning people detectors for tracking in crowded scenes," in *2013 IEEE International Conference on Computer Vision*, pp. 1049–1056, 2013.

[55] Y. Xu, X. Liu, Y. Liu, and S.-C. Zhu, "Multi-view people tracking via hierarchical trajectory composition," in *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 4256–4265, 2016.

[56] K. Bernardin and R. Stiefelhagen, "Evaluating multiple object tracking performance: the clear mot metrics," *EURASIP Journal on Image and Video Processing*, vol. 2008, no. 1, pp. 1–10, 2008.

[57] Y. Li, C. Huang, and R. Nevatia, "Learning to associate: Hybridboosted multi-target tracker for crowded scene," in *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 2953–2960, IEEE, 2009.

[58] E. Ristani, F. Solera, R. S. Zou, R. Cucchiara, and C. Tomasi, "Performance measures and a data set for multi-target, multi-camera tracking," *CoRR*, vol. abs/1609.01775, 2016.

[59] A. Milan and et al., "Mot16: A benchmark for multi-object tracking," *arXiv preprint arXiv:1603.00831*, 2016.

[60] J. I. K. dan Informasi, "Comparison of fairmot-vgg16 and mcmot implementation for multi-object tracking and gender detection on mall cctv," 2021.

[61] J. Berclaz, F. Fleuret, E. Turetken, and P. Fua, "Multiple object tracking using k-shortest paths optimization," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 33, pp. 1806–1819, Sep 2011.

[62] 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.

[63] P. Baqué, F. Fleuret, and P. Fua, "Deep occlusion reasoning for multi-camera multi-target detection," *CoRR*, vol. abs/1704.05775, 2017.

[64] S.-E. Lee, K. Shibata, S. Nonaka, S. Nobuhara, and K. Nishino, "Extrinsic camera calibration from a moving person," *IEEE Robotics and Automation Letters*, vol. 7, no. 4, pp. 10344–10351, 2022.